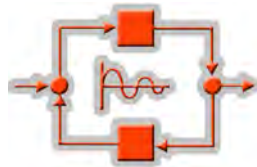




**The International Congress for
global Science and Technology**



**ICGST International Journal on Automatic
Control & System Engineering
(ACSE)**

**Volume (17), Issue (I)
June 2017**

**www.icgst.com
www.icgst-amc.com
www.icgst-ees.com**

**© ICGST LLC, 2017
Delaware, USA**

ACSE Journal
ISSN: Print 1687-4811
ISSN Online 1687-482X
ISSN CD-ROM 1687-4838
© ICGST LLC, 2017, Delaware, USA



Table of Contents

Papers	Pages
P1111701535, author="T.Shravani and N.Krishnakumari", title="Direct Torque Control of Permanent Magnet Synchronous Motor with Three Level Space Vector Modulation",	1--5
P1111707547, author="Prof. Dr. Kamen M. Yanev", title="Digital Optimal Compensation of Process Control Systems",	7--18
P1131711550, author="J. Abliter and E. Agustin and M. Arroyo and A. Morante and J. Taturan and R. Tolentino", title="Employing Vector Multiplication and Tilt Compensation Algorithm in Mimicking a Three Degree of Freedom Robotic Shoulder",	19--26
P1131712557, author="C. Todd and H. Koujan and S. Fasciani", title="A Hybrid Controller for Inflight Stability and Maneuverability of an Unmanned Aerial Vehicle in Indoor Terrains ",	27--39
P1111713560, author="Ali Alqudaihi and Mohamed Zohdy and Hoda Abdel-Aty-Zohdy", title="Applying Hyper-Fuzzy Extended Kalman Filter to Indoor Security Monitoring",	41--51



**ICGST International Journal on Automatic Control & System Engineering -
(ACSE)**

**A Publication of the International Congress for global Science and Technology -
(ICGST)**

ICGST Editor in Chief: Dr. rer. nat. Ashraf Aboshosha

www.icgst.com, www.icgst-amc.com, www.icgst-ees.com

editor@icgst.com



Direct Torque Control of Permanent Magnet Synchronous Motor with Three Level Space Vector Modulation

T.Shravani¹, N.KrishnaKumari²

M.Tech Power Electronics Student, EEE Department, VNR VJIET, Hyderabad, Telangana, India

Assoc.Prof, EEE Department, VNR VJIET, Hyderabad, Telangana, India

shravanithumma@gmail.com, nkkpsg@gmail.com

Abstract

In this paper, the performance of a permanent magnet synchronous motor and impact of the direct torque control with three level space vector modulation is presented. The proposed control method can reduce the flux linkage and torque ripple in a large extent and have a better dynamic and static performance. Space Vector Modulation (SVM) Technique has become the important PWM technique for three phase Voltage Source Inverters for the control of Induction, Brushless DC, Switched Reluctance and Permanent Magnet Synchronous Motors. The study of space vector modulation technique reveals that space vector modulation technique utilizes the DC bus voltage more efficiently and generates less harmonic distortion when compared with Sinusoidal PWM (SPWM) technique. Both SPWM and SVM control strategies are implemented to obtain three level output voltages. In this work, DCMLIs for PMSM with DTC is carried out in MATLAB/Simulink/SimPower System environment. Comparative analysis is investigated for total harmonic distortion (THD) using three level with SVM and SPWM control strategies.

Keywords: PMSM, DTC, SPWM, SVM, MATLAB/SIMULINK.

Nomenclature

V_d	direct axis stator voltage, V
V_q	quadrature axis stator voltage, V
I_d	direct axis stator current, A
I_q	quadrature axis stator current, A
R_d	direct axis resistance, Ω
R_q	quadrature axis resistance, Ω
P	no. of poles
L_d	direct axis inductances, H
L_q	quadrature axis inductances, H
λ_d	direct axis flux linkage, wb
λ_q	quadrature axis flux linkage, wb
λ_f	magnetic flux linkage, wb
Ω_e	rotor speed in electrical, rpm
Ω_m	mechanical speed, rad/s
J	moment of inertia, kg.m ²

B	Viscous Friction Co-efficient, Nm/rad/s
P	Differential operator
T_e	electromagnetic torque, Nm
T_L	load torque, Nm

Acronyms

DCMLI	Diode Clamped Multi Level Inverter
PMSM	Permanent Magnet Synchronous Motor
DTC	Direct Torque Control
SVM	Space Vector Modulation
SPWM	Sinusoidal Pulse Width Modulation

1. Introduction

Permanent-Magnet Synchronous Machines (PMSMs) are used in the recent years due to their advantages over other ac motors such as high torque-to-current ratio, high power-to-weight ratio, high efficiency, high power factor, low noise, and robustness [1] [2]. PMSM Motor is made up of rare earth and neodymium boron magnets; it has been widely used in high performance variable speed industrial applications. In this motor, Permanent Magnets are placed in the rotor, because of absence of windings in the rotor. Rotor copper losses are zero [3]. PMSM are applied in various industrial fields including aviation, rail traction, robotics, electrical vehicles, and servo applications [4][6][9]. Ideal flux circle generated by three-phase symmetric sinusoidal voltage, use the effective voltage vector generated by different switch patterns of inverter to approximate the standard flux circle [5] [6]. Direct torque control (DTC) and field-oriented control (FOC) are the two most popular methods for high performance permanent-magnet synchronous motor (PMSM) drives. Lack of inner current controllers, rotary coordinate transformation, and pulse width modulation in DTC yields a faster dynamic response and a simpler structure in comparison with the FOC method. Although DTC method has the above advantages, it has some drawbacks among which the high torque and stator flux ripples are the most important, followed by variable switching frequency with respect to the motor speed and the load [7]. Compared with FOC, DTC can manipulate stator flux vector directly so as to produce the desired torque [8]. A PI torque



controller is introduced to calculate the voltage reference, which is implemented by the SVM [9] [10].

The paper is organized as follows: Section 2 revolves around the introduction of multilevel inverters. Section 3 explains control strategies SPWM and SVM. Section 4 explains the modelling of PMSM. Section 5 focuses on the simulation results, here there is a comparative analysis of THD, speed and torque responses of PMSM with SPWM and SVM and also with DTC technique. Section 6 summarises the conclusions drawn from the work with future recommendations.

2. Multilevel Inverters

The three methods used for obtaining higher level inverter output voltage, are Connecting devices in series, using switching devices with higher voltage rating, or adopting multilevel topology. The first method can be used only when the problem of dynamic voltage sharing has been efficiently solved. Otherwise it should be avoided. The second method is complicated but it can be compromised in case of the cost, reliability and availability. The limited device voltage rating is the most effective solution for higher level inverter voltage topologies. One of the multilevel structures that has more preference and is widely used is the Neutral-Point-Clamped multilevel inverter or also known as Diode Clamped multilevel inverter. This structure was first proposed in 1980. Basically, NPC multilevel inverters synthesize the small step of staircase output voltage from several levels of DC capacitor voltages. An m-level NPC inverter consists of (m-1) capacitors on the DC bus, 2(m-1) switching devices per phase and 2(m-2) clamping diodes per phase. The three level output voltage can be obtained by using two DC capacitors. Each capacitor has input DC voltage across it and voltage stress will be limited to one capacitor level through clamping diodes. The number of levels can be extended to a higher level by adding switching devices and with these additions, the inverter will be able to achieve higher level step AC voltage, which will be approaching near sinusoidal with minimum harmonics distortion. During inverter operation, the switches near the centre tap capacitors are switched on for a longer period compared to the switches further away from the centre tap. As the switch is further away from the centre tap the switching time is shorter. A multilevel converter has several advantages over a conventional two-level converter that uses high switching frequency pulse width modulation (PWM). Another difference between the conventional 2-level and multilevel NPC is the clamping diode. In case of 3-level NPC inverter, two clamped diodes having the DC bus voltage generates three voltage level, $+V_{dc}/2$, 0 and $-V_{dc}/2$. The attractive features of multilevel inverters can be briefly summarized as follows: i) It produces output voltage with less distortion and here less dv/dt stresses hence reduces electromagnetic problems. ii). Small common mode voltage, stress occurs in the bearings of a motor which MLI reduces to a significant extent. iii). It draws input current with low distortion factor iv). Operation with both fundamental switching frequency and high switching frequency gives less switching loss and higher efficiency.

3. Control Strategies

Modulation techniques and control strategies have been developed for multilevel converters such as sinusoidal pulse width modulation (SPWM), selective harmonic elimination (SHE-PWM), space vector modulation (SVM), and others. The proposed DCMLIs are developed three level with SPWM and SVM.

Space Vector Modulation

Space vector modulation gives fast dynamic response and wide linear range of fundamental voltage compared with the conventional pulse width modulation techniques. The proposed work is based on SVM for multilevel inverter. The SVPWM for multilevel inverters involves mapping of the outer sectors to an inner sub-hexagon sector. This will be very complex, as a large number of sectors and inverter vectors are involved. Conventional space vector PWM (SVPWM) is a widely used PWM strategy which has the advantages of low harmonic distortion in the output current and suitable for digital implementation. In SVPWM technique, actual switching times can be produced by the recombination of effective voltage vectors using the information of the reference voltage vectors location. The flowchart for three level svm DCMLI is depicted in Figure 2.

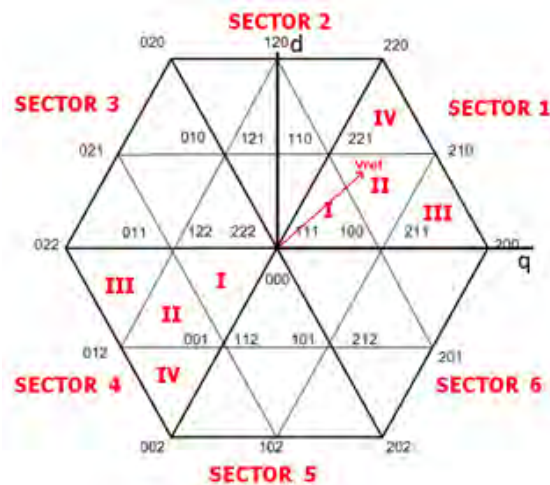


Figure1. Space vector hexagon for three level SVM

4. Direct Torque Control of PMSM

The control methods used for the permanent magnet synchronous motors are: V/f control, field oriented control and direct torque control. Direct Torque control is one of the control methods of the PMSM. In the proposed work, direct torque control of PMSM with three level space vector modulation is considered. In this work PMSM is modeled in MATLAB/SIMULINK environment. Modeling equations for PMSM are given in equations (1) to (8). The proposed block diagram of the work is given in Figure 2.



$$V_d = R_d i_d - \omega_r \Psi_q + \frac{d\Psi_d}{dt} \quad (1)$$

$$V_q = R_q i_q + \omega_r \Psi_d + \frac{d\Psi_q}{dt} \quad (2)$$

$$\Psi_d = L_d i_d + \Psi_{af} \quad (3)$$

$$\Psi_q = L_q i_q \quad (4)$$

Where all voltages (v) and currents (i) refer to the rotor reference frame. The subscripts q_s , d_s , q_r and d_r correspond to q and d axis quantities for the stator(s) and rotor(r) in all combinations, R_a denotes the armature resistance, L_{qs} denotes quadrature axis inductance, L_{ds} denotes direct axis inductance etc.

Then the flux linkages are written as

$$\Psi_d = L_d \dot{i}_d + \Psi_{af} \quad (5)$$

$$\Psi_q = L_q i_q \quad (6)$$

Where L_m is the mutual inductance of the stator winding and rotor magnets.

The developed torque motor is being given by

$$T_e = \frac{3}{2}P(\Psi_d i_d - \Psi_q i_q) \quad (7)$$

The mechanical Torque T_e written in eq.

$$T_e = T_L + B\omega_m + J \frac{d\omega_m}{dt} \quad (8)$$

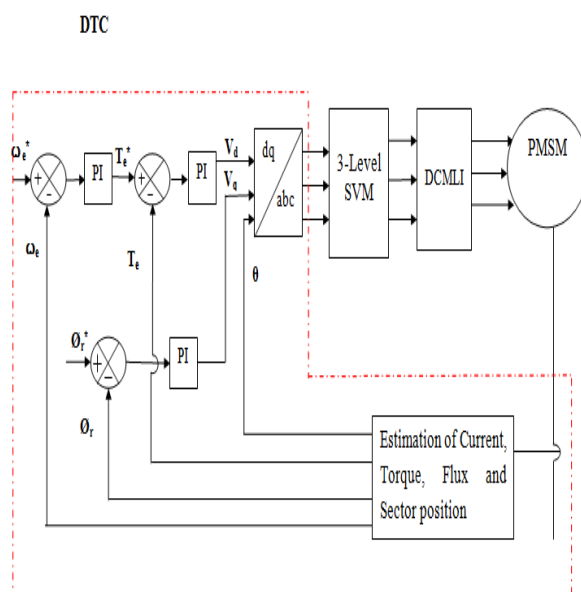


Figure.2. Proposed block diagram

5. Results and Discussion

The dynamic performance of direct torque control permanent magnet synchronous motor using three level space vector control is carried out using MATLAB/Simulink.

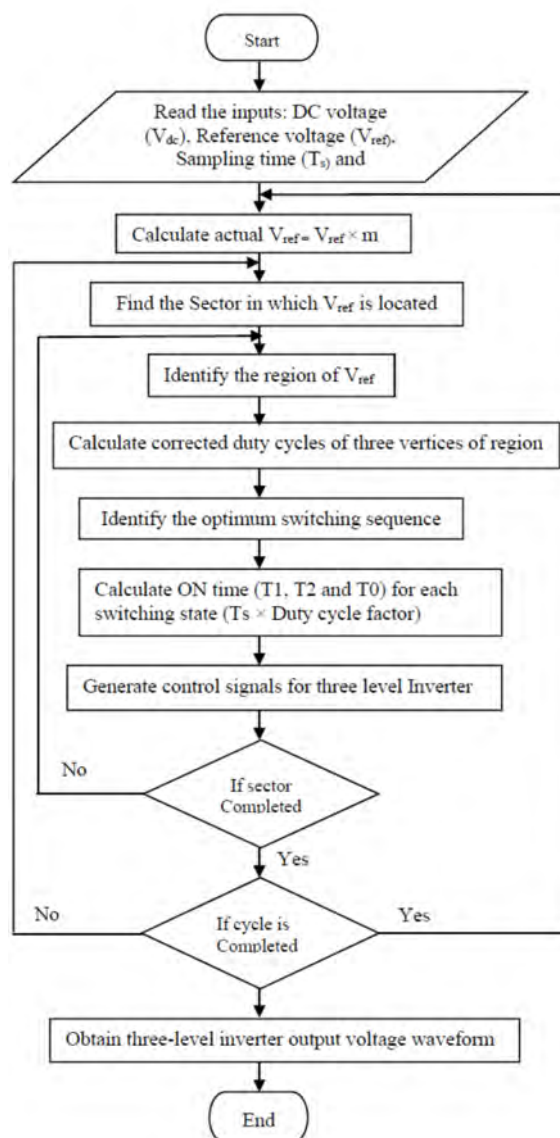


Figure3. The flow chart for three level SVM

There are three test cases; at first the analysis of Three Level DCI Fed PMSM is carried out Using SPWM. In second case the simulation is carried out for the Three Level DCI Fed PMSM Using SVM. Finally the dynamic analysis is carried out for DTC of PMSM with three level SVM. The results are plotted from Figure 4 to Figure 6. In all three cases the load torque 10Nm is applied at $t=1$ sec. In the case of DTC with SVM gives better results with reduced harmonics. Torque ripples and steady state error also got reduced to low value.

The results depict that as the level of phase voltage increases, the THD decreases for phase voltage and currents, also the torque ripples reduces, which is clearly shown from Figure 4 with SPWM, Figure 5 with SVM and from Figure 6 with DTC with three level space vector modulation. In these three cases DTC with SVM gives better results with reduced harmonics, torque ripples and



steady state error also got reduced to low value. The THD analysis is given in Table 1. It is very well clear that THD for phase voltage reduces from 37.61% to 3.77% and current reduces from 4.27% to 2.53% respectively. The results have been presented and analysed in this paper. The parameters of the PMSM and dc voltage considered for this paper is presented in Table 2.

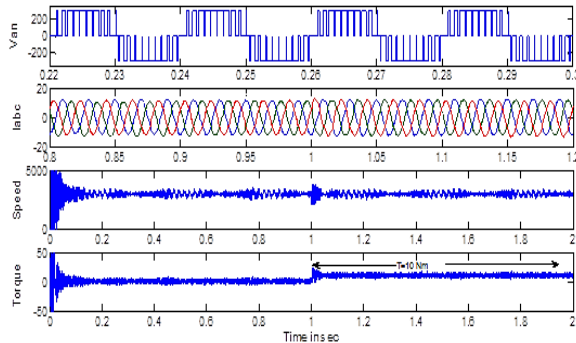


Figure 4. Three Level DCI Fed PMSM Using SPWM when load torque 10 Nm applied at t=1 sec

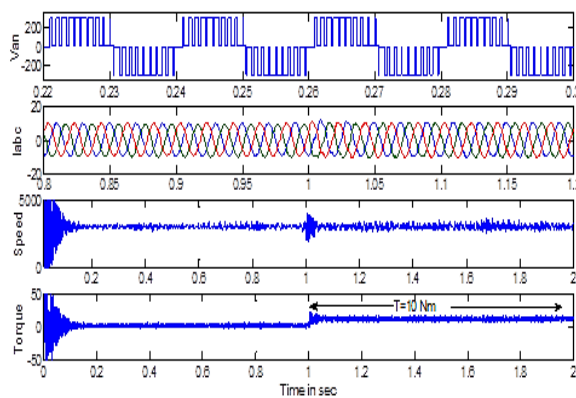


Figure 5. Three Level DCI Fed PMSM Using SVM when load torque 10 Nm applied at t=1 sec

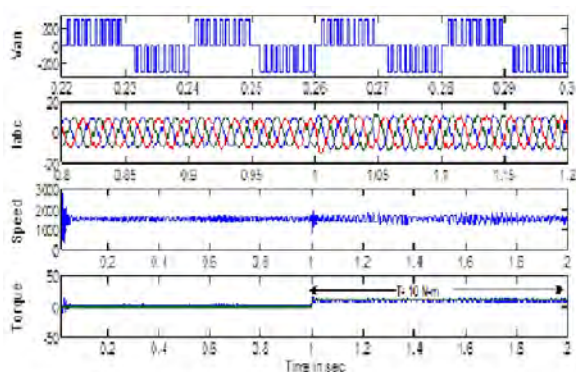


Figure 6. Three Level DCI Fed PMSM with DTC Using SVM when load torque 10 Nm applied at t=1 sec

Table 1. THD Analysis for 3 Level Inverter fed PMSM

S. No	V_{THD}	I_{THD}
SPWM		
1	37.61%	4.27%
SVM		
2	3.77%	3.05%
DTC-SVM		
3	3.77%	2.53%

Table2. PMSM Parameters

Parameter	Description Value
Stator Resistance	2.875 Ω
Direct Axis Inductance	0.0085 H
Quadrature Axis Inductance	0.0085 H
Moment of Inertia	0.0008 Kg m ²
Friction Viscous Gain	0 Nm/rad/s
Number of Poles	8
DC Voltage	600 V

6. Conclusion

The inverter output voltage is proportional to the number of active switches and the gate signals generated using SVM gives low harmonic distortion. In this paper three level with SPWM and SVM is implemented for DTC of PMSM. By using SVM, the dynamic and static performance of the control system gives better performance using DTC of PMSM. Simulation results indicate that the proposed SVM-DTC can reduce the flux and torque ripple efficiently. It is further suggested that the proposed work further can be implemented with fuzzy logic and model reference adaptive control techniques.

7. References

- [1] M. Nasir Uddin, HonBin Zou, and F. Azevedo, "Online Loss-Minimization-Based Adaptive Flux Observer for Direct Torque and Flux Control of PMSM Drive", IEEE ,Transactions On Industry Applications, Vol. 52, 2016.
- [2] Mohammad Hossein Vafaie, Behzad Mirzaeian Dehkordi, Payman Moallem, and Arash Kiyomarsi, "A New Predictive Direct Torque Control Method for Improving Both Steady-State and Transient-State Operations of the PMSM", IEEE ,Transactions On Power Electronics, Vol. 31, 2016.
- [3] H. Bouzeria, C. Fetha, T. Bahi, I. Abadlia, Z. Layate, S. Lekhchine, "Fuzzy Logic Space Vector Direct Torque Control of PMSM for Photovoltaic Water Pumping System", Elsevier , Energy Procedia Vol 74, 2015, pages 760 – 771.
- [4] Badreddine Naas, L. Nezli, Bachir Naas, M.O. Mahmoudi, M.Elbar, "Direct Torque Control Based Three Level Inverter-fed Double ",



Elsevier, Energy Procedia , Vol 18 , 2012, pages 521 – 530.

- [5] Xu Wang, Yan Xing, Zhipeng He, Yan Liu, “Research and Simulation of DTC Based on SVPWM of PMSM”, Elsevier, Procedia Engineering Vol 29, 2012, pages 1685 – 1689.
- [6] Lachtar Salah, Bahi Tahar, “SVPWM performance of PMSM variable speed and impact of diagnosis sensors faults”, Elsevier, Energy Procedia Vol 74, 2015, pages 679 – 689.
- [7] Mohammad-Hossein Vafaie, Behzad Mirzaeian Dehkordi, Payman Moallem, Arash Kiyomarsi, “Improving the Steady-State and Transient-State Performances of PMSM through an Advanced Deadbeat Direct Torque and Flux Control System”, IEEE, Transactions on Power Electronics, Vol 9, 2015.
- [8] Changliang Xia, Shuai Wang, Xin Gu, Yan Yan, and Tingna Shi, “Direct Torque Control for VSI-PMSM Using Vector Evaluation Factor Table”, IEEE, Transactions On Industrial Electronics, Vol. 63, 2016.
- [9] Qian Liu and Kay Hameyer, “Torque Ripple Minimization for Direct Torque Control of PMSM With Modified FCSMPC”, IEEE, Transactions On Industry Applications, Vol. 52, 2016.
- [10] O. Ouledali, A .Meroufel, P. Wira ,S.Bentouba, “ Direct torque fuzzy control of PMSM based on SVM ”, Elsevier, Energy Procedia Vol 74, 2015, pages 1314 – 1322.

Biographies



T. Shravani, received B.Tech degree from JNTUHCEJ affiliated to JNTU Hyderabad, Telangana in the year 2013. Pursuing M.Tech in Power Electronics in VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad. Her research areas include PMSM, Multi Level Inverters.



N. Krishna Kumari graduated in Electrical and Electronics Engineering from Sri Venkateshwara University, Thirupathi, Andhra Pradesh, India in 1994. Received M.E. in Electrical machines from PSG College of Technology, Coimbatore, Bharathiyar University, Tamilnadu, India in 1997. She submitted her Ph.D thesis in the area of Electrical Drives and Control at Jawaharlal Nehru Technological University, Hyderabad in July 2016, India. She has presented/ published 27 research papers in national and international conferences and journals. Her research areas include PWM techniques, DC-DC converters, Multi Level Inverters, AC- AC converters and control of electrical drives. She executed UGC Minor Project on “FPGA Implementation of Field Oriented Control for Permanent Magnet Synchronous Motor”, in September, 2014. She is a IEEE Member, Life Member of ISTE and SAE. She is presently working as Associate Professor in the Electrical and Electronics Engineering Department, VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, India.







Digital Optimal Compensation of Process Control Systems

Kamen M. Yanev¹

Department of Electrical Engineering, University of Botswana, Gaborone, Botswana

yanevkm@yahoo.com, yanevkm@mopipi.ub.bw

http://www.ub.bw

Abstract

This paper is suggesting a practical method for realizing optimal digital compensation of process control systems. Initially, a cascade continuous compensator prototype is designed and further it is converted into its digital equivalent. The purpose of the compensator is to eliminate some properly selected poles of the original control system transfer function, to introduce new dominant poles and specific system amplification. As a result, the system's transient response and stability are considerably improved. The control system's relative damping ratio, percent maximum overshoot, phase margin, rise and settling times, are optimized.

Keywords: Marginal system; Dominant poles; Stability; Analogue prototype; Lag-Lead multistage compensation; Digital system; Optimal system; Transient responses;

Nomenclature

ITAE	Integral of the Time-weighted Absolute Error
ζ (DR)	Closed-loop relative damping ratio
PMO	Percent Maximum Overshoot
t_s	Settling time
t_m	Maximum overshoot time
e_{ss}	Steady-State error
$G_{PI}(s)$	Transfer function of the plant open-loop system
s	Laplace operator
K, K'	Variable gains of the linear system
$G_P(s)$	Characteristic equation of the open-loop system
$G_c(s)$	Transfer function of the analogue compensator
$G_c'(s)$	Transfer function of the multi-stage lead section of the compensator continuous prototype
$G_c''(s)$	Transfer function of the attenuation
$G_c'''(s)$	Transfer function of the of the lag compensation and amplification
T_s	Sampling period
T_{min}	Minimum time-constant of the linear section of the system
ω_s	Sampling frequency
z	z-transform operator
$G_{cd}(z)$	Transfer function of the series compensation controller stage in the discrete-time domain

1. Introduction

High quality performance of control systems can be achieved by applying the suggested cascade optimal digital compensation. It is implemented by employing a number of steps for the design of a proper digital compensator. The purpose of the compensator is to eliminate some properly selected poles of the system's transfer function. At the same time it introduces new dominant poles and a specifically designed amplification. As a result, the quality of the system's performance in terms of its transient response, stability and accuracy are considerably improved. The suggested technique of successive steps brings the system to its marginal state, further subjecting it to multi-stage phase-lead, phase-lag compensation and amplification. The continuous cascade compensator prototype is finally converted into its digital equivalent and implemented as a microcontroller compensator. The main purpose of the suggested method is the compensated control system to meet the ITAE criterion and accordingly, the following objectives [1]:

$$\zeta = 0.707; (PMO) \leq 4\%; t_s/t_m \leq 1.4; e_{ss} \leq 1\% \text{ (type 0)}$$

Phase-lead and phase-lag compensation technique is a well known mechanism for improving control systems' performance as demonstrated in considerable number of sources [1], [2], [3], [4]. The disadvantage of this method of compensation, as described in the sources, is that it only recommends the range of the compensator's frequencies and it depends on the designer's expertise to achieve the best performance of the compensated system. Further, the direct application of the compensation technique causes compromise with the system's gain.

Another method to meet the objectives of the ITAE criterion is to apply a PI or a PID controller to process control systems. This is a well known approach and it is also described in the sources mentioned above. Different methodologies are recommended for determination of the PI or PID controller constants where results are usually achieved with a lot of assumptions. Finally, the best system performance is proved to be accomplished by a procedure of manual MATLAB interaction tuning of the PI or PID controller constants [5].



The importance of the new controller design technique, presented by the author of this research, is that it is suggesting a straight forward and exact achievement of the ITAE criterion objectives. Further contribution of the suggested by the author technique is the conversion of the achieved controller from its continuous to its digital equivalent. In this way the compensation is implemented with the aid of microcontrollers that are programmed based of the difference equations, representing the digital transfer functions of the controller stages. The application of digital control is in accordance to the Euler's approximation [6], [7], [8]. It recommends that the sampling period T_s of the discrete system, should be within the range $T_s \leq (0.1T_{min} \text{ to } 0.2T_{min})$, where T_{min} is the minimum time-constant of the continuous-time system or the analogue plant model prototype. Only then there will be a close match between the discrete and the continuous-time system performance and both, the optimal plant analogue prototype and its optimal digital equivalent are having exactly the same relative damping of $\zeta = 0.707$. If the Euler's approximation is applied, it is apparent that there is a complete match between the corresponding pole location on the s-plane and on the z-plane. Due to this conclusion, the controller stages are designed in the continuous-time domain and further converted into their discrete-time equivalents [9].

For the clarity of the presented research, this paper is organized in the following sequence. Initially, the compensation design for a system Type 0 is described for the cases of considerable difference and insignificant difference between dominant and insignificant system poles. With the aid of three successive steps the system's performance is preliminary evaluated. Further, the implementation of five successive rules are bringing the system's performance to perfection, satisfying the ITAE criterion objectives. This is followed by the application of the compensation technique to a system Type 1. Finally, a pseudo code technique is applied for describing the different system cases with the aid of a flowchart, in this way clarifying the paths and steps that are to be followed for the design of the compensation.

2. Compensator Design for a System Type 0

Case 1: Considerable Difference between Dominant and Insignificant Poles

The compensation design can be demonstrated for a case of a **cruise control system** [10] with a transfer function:

$$\begin{aligned} G_{p1}(s) &= \frac{K'}{(1+T_1s)(1+T_2s)(1+T_3s)} = \\ &= \frac{10}{(1+0.1s)(1+0.2s)(1+s)} = \\ &= \frac{500}{s^3 + 16s^2 + 65s + 50} = \frac{K}{s^3 + 16s^2 + 65s + 50} \end{aligned} \quad (1)$$

Step 1: Evaluation of the Original Control System

The system's performance is unacceptable due to large oscillations as seen from Figure 1. The system's transient response is achieved by applying the following MATLAB code:

```
>> Gp1=tf([0 500],[1 16 65 50])
>> Gfb1=feedback(Gp1,1)
>> step(Gfb1)
```

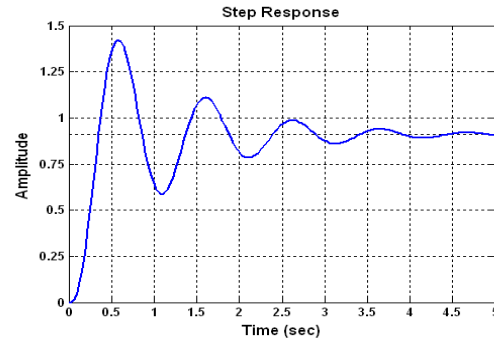


Figure 1: Transient Response of the Original Control System

Step 2: Determination of the Marginal System Gain

To apply the suggested compensation technique, the system initially has to be brought to its marginal state by tuning its original gain K' . The critical value of K' is achieved by means of applying the proposed by the author, in some of his previous publications, method of Advanced D-Partitioning [11], [12], [13].

To apply this method, the characteristic equation of the system (2) is derived from Equation (1), considering the system's gain value $K' = 0.02K$ as unknown.

$$G_P(s) = s^3 + 16s^2 + 65s + 50 + K \quad (2)$$

From Equation (2) the unknown value of K can be presented as follows:

$$K = -s^3 - 16s^2 - 65s - 50 \quad (3)$$

The D-partitioning curve in terms of the variable parameter K can be plotted in the complex plane within the frequency range $-\infty \leq \omega \leq +\infty$ facilitated by MATLAB the "nyquist" m-code, as seen in Figure 2.

```
>> K=tf([-1 -16 -65 -50],[0 1])
Transfer function:
-s^3 - 16 s^2 - 65 s - 50
>> nyquist(K)
```

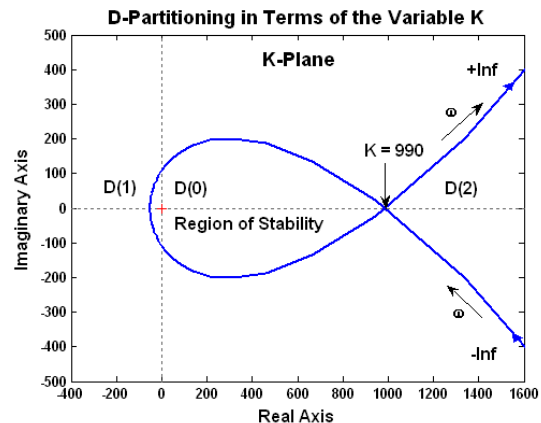


Figure 2: D-Partitioning Facilitated by the "nyquist" m-code



To avoid any misinterpretation of the D-Partitioning procedure, the “nyquist” m-code can be modified into a “dpartition” m-code with the aid of the MATLAB Editor and a proper formatting.

It is seen that the D-partitioning determines three regions on the K-plane: D(0), D(1) and D(2). Applying the Advanced D-Partitioning method, only D(0) is the region of stability, being always on the left-hand side of the curve for the frequency range from $-\infty$ to $+\infty$ [11], [12].

At $K = 990$, the system is marginal and the system's gain becomes $K' = 0.02K = 0.02 \times 990 = 19.8$.

Comparison between the original and the marginal system transient responses is shown in Figure 3.

```
>> Gp1=tf([0 500],[1 16 65 50])
>> Gfb1=feedback(Gp1,1)
>> Gp2=tf([0 990],[1 16 65 50])
>> Gfb2=feedback(Gp2,1)
>> step(Gfb1,Gfb2)
```

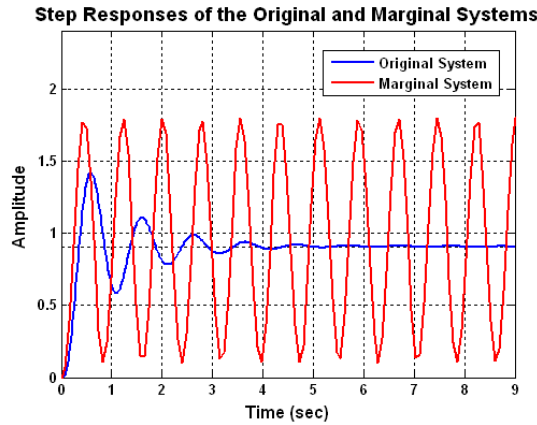


Figure 3: Comparison between the Original and the Marginal Linear Prototype Systems' Transient Responses

Considering Equation (1), the transfer function of the marginal control system is presented as:

$$G_{P2}(s) = \frac{19.8}{(1 + 0.1s)(1 + 0.2s)(1 + s)} \quad (4)$$

Step 3: Optimization of ζ , t_s/t_m and PMO

Initially, the cascade compensator is designed as a continuous prototype and further it is converted into its digital equivalent. The following rules are applied:

Rule 1: To optimize ζ , t_s/t_m and the PMO of a Type 0 marginal closed-loop system, a cascade multi-stage compensation with factors of $\alpha_{1,2,3,\dots} = 10$ is applied for a zero-pole cancellation.

This rule is considered from the graphical relationship of Figure 4 [13], [14].

The number of the compensating stages N is to be one less than the order of the open-loop system, i.e. $N = n + m - 1$.

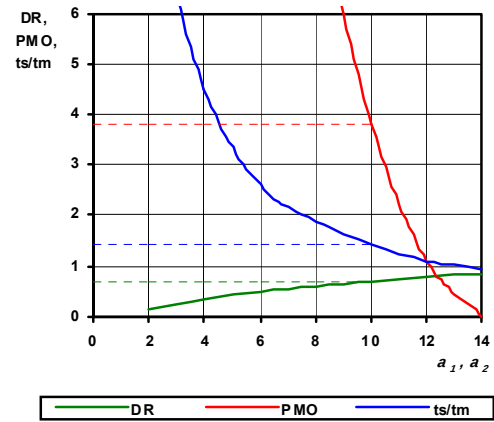


Figure 4: Determination of the Optimum Values of α_1 and α_2

The most dominant pole of the open-loop system should be left uncompensated. Since the plant transfer function from Equation (1) is of a third order, according to Rule 1, the multi-stage lead section of the compensator continuous prototype should have a transfer function:

$$\begin{aligned} G'_c(s) &= \frac{(1 + \alpha_1 T_1 s)(1 + \alpha_2 T_2 s)}{\alpha_1 \alpha_2 (1 + T_1 s)(1 + T_2 s)} = \\ &= \frac{(1 + 0.1s)(1 + 0.2s)}{100(1 + 0.01s)(1 + 0.02s)} \end{aligned} \quad (5)$$

Rule 2: The current gain is maintained by amplification equal to the product $\alpha_1 \alpha_2 \alpha_3 \dots$ [13], [14].

$$G''_c(s) = \alpha_1 \alpha_2 = 10 \times 10 = 100 \quad (6)$$

Rule 3: To optimize a marginal closed-loop system, single-stage section of the continuous prototype lag compensator with factor $\beta = 10$ is to be applied for a zero-pole cancellation of the uncompensated most dominant pole of the open-loop system.

To optimize the steady-state error e_{ss} the current gain should be increased by the factor $\gamma = 10$ (proved from Figure 5 [13], [14]). Both relationships shown in Figure 4 and Figure 5 are achieved by tracking procedures.

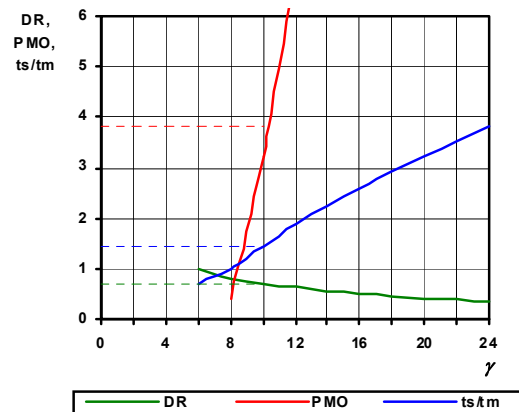


Figure 5: Determination of Optimum Value of γ



The most significant pole in Equation (1) is $p_3 = -1$. Applying Rule 2, the section of the lag compensation and amplification is presented by:

$$G_c'''(s) = \frac{\gamma(1+T_3s)}{(1+\beta T_3s)} = \frac{10(1+s)}{(1+10s)} \quad (7)$$

The transfer function of the full compensator is the product of the transfer functions of the three sections:

$$G_c(s) = G_c'(s) \times G_c''(s) \times G_c'''(s) = \frac{10(1+0.1s)(1+0.2s)(1+s)}{(1+0.01s)(1+0.02s)(1+10s)} \quad (8)$$

Applying the full compensation, taking into account equations (4) and (8), the transfer function of the open-loop system becomes:

$$\begin{aligned} G(s) &= G_c(s) \times G_p(s) = \\ &= \frac{100}{(1+0.01s)(1+0.02s)(1+10s)} \\ &= \frac{100}{0.002s^3 + 0.3002s^2 + 10.03s + 1} \end{aligned} \quad (9)$$

Comparison between the transient responses of the original, the marginal and the compensated system, as shown in Figure 6, is achieved by the code:

```
>> Gp1=tf([0 500],[1 16 65 50])
>> Gfb1=feedback(Gp1,1)
Transfer function:
      500
-----
s^3 + 16 s^2 + 65 s + 550
>> Gp2=tf([0 990],[1 16 65 50])
>> Gfb2=feedback(Gp2,1)
Transfer function:
      990
-----
s^3 + 16 s^2 + 65 s + 1040
>> Gp3=tf([0 100],[0.002 0.3002 10.03 1])
>> Gfb3=feedback(Gp3,1)
Transfer function:
      100
-----
0.002 s^3 + 0.3002 s^2 + 10.03 s + 101
>> step(Gfb1,Gfb2,Gfb3)
```

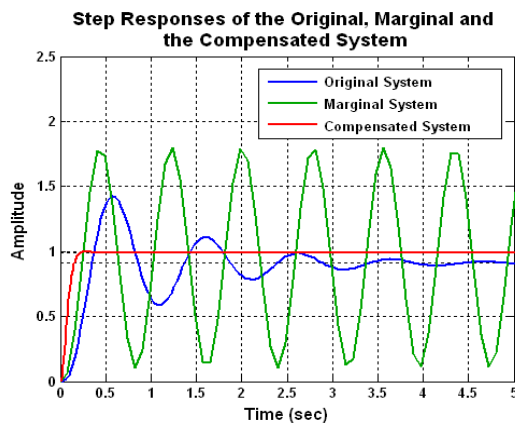


Figure 6: Comparison between the Transient Responses of the Original, Marginal and Compensated Linear Prototype Systems

The Bode diagram of the compensated system is plotted by applying the code:

```
>> Gp3=tf([0 100],[0.002 0.3002 10.03 1])
>> margin(Gp3)
```

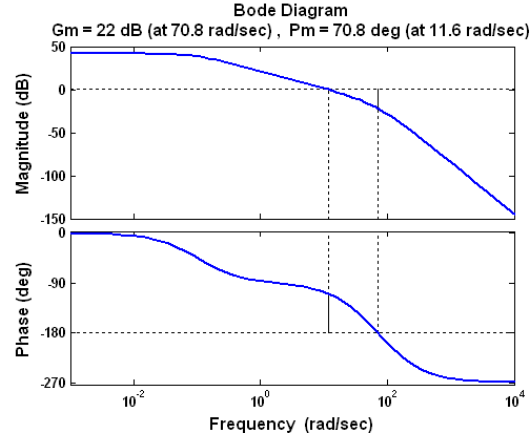


Figure 7: Bode Diagrams of the Compensated System Proving Phase Margin PM = 70.8 and Damping $\zeta \approx 0.707$

The system performance after the compensation, seen from the transient response in Figure 6 and the Bode diagram in Figure 7, demonstrates the achievement of optimal performance [1], [2], [14].

Step 4: Conversion of the Continuous Prototype Compensated System into its Digital Equivalent.

In order to apply digital compensation control, in accordance to the Euler's approximation [15] the sampling period T_s of the discrete system, should be within the range $T_s \leq (0.1T_{min} \text{ to } 0.2T_{min})$, where T_{min} is the minimum time-constant of the continuous-time system.

By applying the Euler's approximation, a close match is expected between the discrete and the continuous-time system performance. For the case under discussion, the plant's minimum time constant is $T_{min} = 0.1\text{sec}$ and the sampling period is chosen as $T_s = 0.01\text{ sec}$, satisfying the condition $T_s \leq 0.1T_{min}$. When analysis is applied in the discrete-time domain, the considered frequency variation is within the range $\omega = \pm \omega_s/2 = \pm 2\pi/2T_s$ [6], [16].

The analysis in the discrete-time domain is applied with the aid of the Bilinear Tustin Transform [16], [17] that is a first-order approximation of the natural logarithm function, being an exact mapping of the z-plane to the s-plane and vice versa. It maps positions from the $j\omega$ axis in the s-plane to the unit circle $|z| = 1$ in the z-plane. The Bilinear Tustin approximation and its inverse are presented as [16]:

$$z = e^{sT_s} = \frac{e^{sT_s/2}}{e^{-sT_s/2}} \approx \frac{1+sT_s/2}{1-sT_s/2}; s = \frac{1}{T_s} \ln(z) \approx \frac{2}{T_s} \frac{z-1}{z+1} \quad (10)$$

To employ the Bilinear Tustin Transform, the MATLAB 'tustin' code is to be applied, performing the transform of the continuous-time to discrete-time system equivalents. The results demonstrated in Figure 6 and Figure 7 can be compared with the results after converting the analogue system model prototype into its digital equivalent:




```

>> Gp1=tf([0 500],[1 16 65 50])
>> Gfb1=feedback(Gp1,1)
>> Gp3=tf([0 100],[0.002 0.3002 10.03 1])
>> Gfb1d = c2d(Gfb1,0.01,'tustin')
Transfer function:
5.778e-005 z^3 + 0.0001733 z^2 + 0.0001733 z + 5.778e-005
-----
z^3 - 2.846 z^2 + 2.698 z - 0.852
Sampling time: 0.01
>> Gfb3d = c2d(Gfb3,0.01,'tustin')
Transfer function:
0.003321 z^3 + 0.009962 z^2 + 0.009962 z + 0.003321
-----
z^3 - 1.916 z^2 + 1.139 z - 0.1958
Sampling time: 0.01
>> step(Gfb1d,Gfb3d)

```

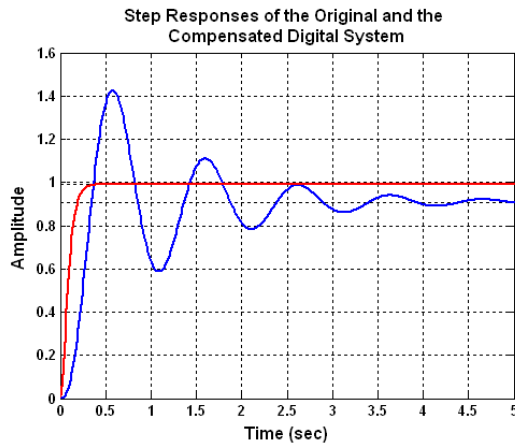


Figure 8: Comparison between the Transient Responses of the Original and Compensated Digital Systems

The Bode diagram of the compensated digital system is plotted by applying the code:

```

>> Gp3=tf([0 100],[0.002 0.3002 10.03 1])
>> Gp3d = c2d(Gp3,0.01,'tustin')
Transfer function:
0.003332 z^3 + 0.009995 z^2 + 0.009995 z + 0.003332
-----
z^3 - 1.932 z^2 + 1.132 z - 0.1998
Sampling time: 0.01
>> margin(Gp3d)

```

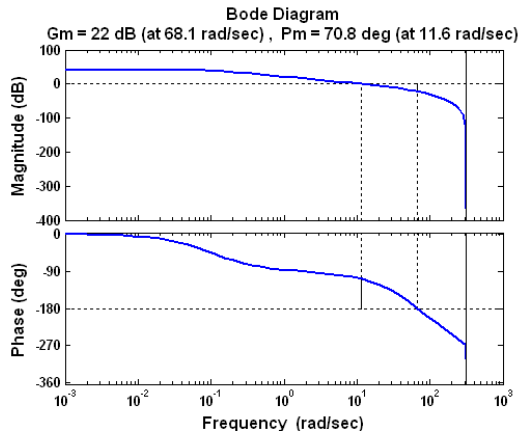


Figure 9: Bode Diagrams of the Digital Compensated System Proving Phase Margin PM = 70.8 and Damping $\zeta \approx 0.707$

The results obtained from Figure 8 and Figure 9, compared with the results from Figure 6 and Figure 7, prove the close match between the performance of the continuous-time model prototype and the digital model of the system (PM = 70.8 and $\zeta \approx 0.707$).

The performance specifications of the compensated system, determined from the transient response and the Bode diagram shown in Figure 8 and Figure 9, are compared with the objectives for optimal performance.

Table 1
Objectives and Real Results for Case 1

Specifications	Objectives	Real Results	Consideration
ζ	= 0.707	= 0.707	Matching
PMO	$\leq 4\%$	= 2.8%	Better
$t_{s(1\%)/t_m}$	≤ 1.49	= 1.25	Better
$e_{ss}(t)$	< 1%	= 0.06%	Better

From the graphs at Figure 8 and Figure 9 and from the summary in Table 1, it is seen that the transient response of the compensated system in terms of relative damping ratio ζ , percent maximum overshoot (PMO), time ratio $t_{s(1\%)/t_m}$ and steady-state error $e_{ss}(t)$ is either matching or it is better than the one of the set objective.

Step 5: Design of a Digital Compensation Controller based on a Microcontroller (Case 1).

The design of a digital compensation controller founded on its analogue prototype is realized again by applying the Tustin bilinear transform [16], [17]. From the Equation (8), the transfer function of the modified compensation controller stage can be presented as:

$$G_c(s) = \frac{10(1+0.1s)(1+0.2s)(1+s)}{(1+0.01s)(1+0.02s)(1+10s)} = \frac{2s^3 + 32s^2 + 130s + 100}{0.02s^3 + 3.002s^2 + 100.3s + 10} \quad (11)$$

The digital equivalent of the series compensator controller stage [16], [17] is realized by the code:

```

>> Gc=tf([2 32 130 100],[0.02 3.002 100.3 10])
>> Gcd = c2d(Gc,0.1,'tustin')
Transfer function:
9.328 z^3 - 17.15 z^2 + 9.743 z - 1.688
-----
z^3 + 0.1052 z^2 - 0.7986 z - 0.2829
Sampling time: 0.1

```

Following the code result, the transfer function of the series compensation controller stage in the discrete-time domain is as follows:

$$G_{cd}(z) = \frac{9.328z^3 - 17.15z^2 + 9.743z - 1.688}{z^3 + 0.1052z^2 - 0.7986z - 0.2829} = \frac{Y(z)}{X(z)} \quad (12)$$

To enable the implementation of a microcontroller that is connected in cascade with the original plant, the transfer function of the series digital robust control stage is represented by the following difference equation [18]:



$$\begin{aligned}
y(kT) = & 0.1052 y[(k-1)T] - 0.7986 y[(k-2)T] + \\
& - 0.2829 y[(k-3)T] + \\
& + 9.328 x(kT) - 17.15 x[(k-1)T] + \\
& + 9.743 x[(k-2)T] - 1.688 x[(k-3)T]
\end{aligned} \quad (13)$$

Case 2: Insignificant Difference between Dominant and Insignificant Poles

Rule 4: If the most significant pole of the open-loop system has a real part close in value to that of an insignificant pole, Rule 3 is modified to Rule 4, where successive two-stage lag compensation with factors $\beta_1 = \beta_2$ and a factor δ should be applied.

The values of the factors β_1 and β_2 at Rule 4 are chosen by the same considerations as in Rule 2. The value of the factor δ is varied by a tracking procedure on the transient response, searching for the optimum performance of the compensated system. It is seen from Figure 10 that the set of objectives can be met if the factor $\delta = 80$ [13], [18].

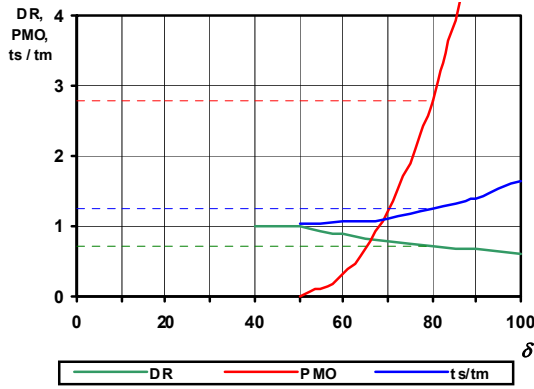


Figure 10: Determination of Optimum Value of δ

Rule 4 can be illustrated for a system Type 0 [19] of an **armature control dc motor and a type driving mechanism** with original gain $K = 10$. Applying the D-partitioning technique it is brought to its marginal state, where its gain becomes $K = 20$, as demonstrated below:

$$G_{P3}(s) = \frac{20}{(1+0.01s)(1+0.02s)(1+0.1s)} \quad (14)$$

The real part of the most significant pole is $p_3 = -10$ and it is only 5 times smaller than the real part of the one of the insignificant poles $p_1 = -50$.

Again, in accordance to Rule 1, a two-stage lead compensation and attenuation is applied, where the factors are $\alpha_1 = \alpha_2 = 10$. Following Rule 2, the gain is maintained by amplification equal to the product $\alpha_1 \alpha_2$. Further, following Rule 4, a two-stage lag compensation with factors $\beta_1 = \beta_2 = 10$ and a factor amplification $\delta = 80$ is suggested. Using a related sequence and applying Equation (5), the two-stage lead stage is presented by:

$$\begin{aligned}
G'_c(s) &= \frac{(1+\alpha_1 T_1 s)(1+\alpha_2 T_2 s)}{\alpha_1 \alpha_2 (1+T_1 s)(1+T_2 s)} = \\
&= \frac{(1+0.01s)(1+0.02s)}{100(1+0.001s)(1+0.002s)}
\end{aligned} \quad (15)$$

Applying Rule 2, the second compensation section is achieving the amplification:

$$G''_c(s) = \alpha_1 \alpha_2 = 10 \times 10 = 100 \quad (16)$$

Applying Rule 4, the two-stage lag compensation and amplification is presented by:

$$G'''_c(s) = \frac{\delta(1+T_3 s)(1+T_4 s)}{(1+\beta_1 T_3 s)(1+\beta_2 T_4 s)} = \frac{80(1+0.1s)(1+s)}{(1+s)(1+10s)} \quad (17)$$

The transfer function of the complete compensator as a product of the three sections is:

$$\begin{aligned}
G_c(s) &= G'_c(s) \times G''_c(s) \times G'''_c(s) = \\
&= \frac{80(1+0.01s)(1+0.02s)(1+0.1s)}{(1+0.001s)(1+0.002s)(1+10s)}
\end{aligned} \quad (18)$$

After applying the full compensation, considering Equations (14) and (18), the analogue prototype of the transfer function of the open-loop compensated system becomes:

$$\begin{aligned}
G(s) &= G_c(s) \times G_p(s) = \\
&= \frac{1600}{(1+0.001s)(1+0.002s)(1+10s)}
\end{aligned} \quad (19)$$

In the same way the Bilinear Tustin Transform is applied, converting the continuous-time system prototype into its digital equivalent. The system's transient response before and after compensation is shown in Figure 11 and is obtained by the following code:

```

>> Gp1=500000/(s+10)/(s+50)/(s+100)
>> Gpfb1=feedback(Gp1,1)
>> G=80000000/(s+0.1)/(s+500)/(s+1000)
>> Gfb=feedback(G,1)
>> Gpfb1d=c2d(Gpfb1,0.002,'tustin')
>> Gfbd=c2d(Gfb,0.002,'tustin')
>> step(Gpfb1d,Gfbd)

```

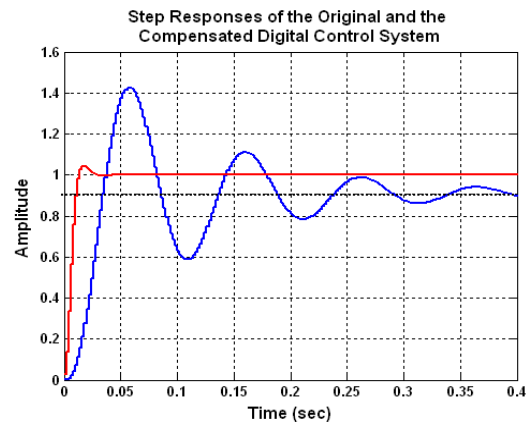


Figure 11: Comparison between the Transient Responses of the Original and Compensated Digital System (Case 2)

The Bode diagram of the compensated digital system is shown at Figure 11 and plotted by applying the code:



```
>> G=80000000/(s+0.1)/(s+500)/(s+1000)
>> Gd = c2d(G,0.002,'tustin')
Zero/pole/gain:
0.026664 (z+1)^3
-----
z (z-1) (z-0.3333)
Sampling time: 0.002
```

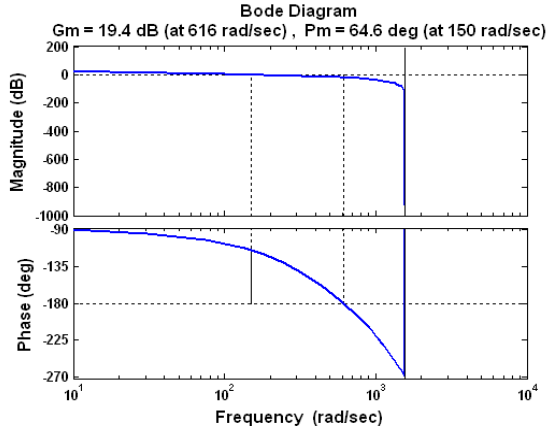


Figure 12: Bode Diagrams of the Digital Compensated System Proving Phase Margin PM = 64.6 and Damping $\zeta \approx 0.646$

From the summary in Table 2, it is obvious that the objectives are met. Again, the real results for the compensated control system are either close or better than the set specifications.

Table 2
Objectives and Real Results for Case 2

Specifications	Objectives	Real Results	Consideration
ζ	= 0.707	= 0.646	Close
PMO	$\leq 4\%$	= 2.8%	Better
$t_{s(1\%)/t_m}$	≤ 1.49	= 1.25	Better
$e_{ss}(t)$	< 1%	= 0.06%	Better

Step 5: Design of a Digital Compensation Controller based on a Microcontroller (Case 2).

The digital equivalent of the series compensator controller stage can be realized in a similar procedure, like the one implemented for example of Case 1. The transfer function compensation controller stage shown in equation (18) is modified and can be presented in the following format:

$$G_c(s) = \frac{80(1+0.01s)(1+0.02s)(1+0.1s)}{(1+0.001s)(1+0.002s)(1+10s)} = \frac{0.0016s^3 + 0.256s^2 + 10.4s + 80}{0.00002s^3 + 0.030002s^2 + 10.003s + 1} \quad (20)$$

Considering the equation (20), the digital equivalent of the series compensator controller stage is realized by the code:

```
>> Gc=tf([0.0016 0.256 10.4 80],[0.00002 0.030002 10.003 1])
Transfer function:
0.0016 s^3 + 0.256 s^2 + 10.4 s + 80
-----
2e-005 s^3 + 0.03 s^2 + 10 s + 1
>> Gcd = c2d(Gc,0.01,'tustin')
Transfer function:
1.891 z^3 + 1.441 z^2 - 0.1501 z - 0.1801
-----
z^3 + 0.8938 z^2 - 0.9782 z - 0.8781
Sampling time: 0.01
```

Following the code result, the transfer function of the series compensation controller stage in the discrete-time domain is as follows:

$$G_{cd}(z) = \frac{1.891z^3 + 1.441z^2 - 0.1501z - 0.1801}{z^3 + 0.8938z^2 - 0.9782z - 0.8781} = \frac{Y(z)}{X(z)} \quad (21)$$

Taking into accounting equation (21), the transfer function of the series digital robust control stage is represented by the following difference equation (22). A microcontroller, connected in cascade with the original plant, can be programmed in accordance with this difference equation [18], [20]:

$$y(kT) = 0.8938y[(k-1)T] - 0.9782y[(k-2)T] + 1.891x(kT) + 1.441x[(k-1)T] - 0.1501x[(k-2)T] - 0.1801x[(k-3)T] \quad (22)$$

3. Compensator Design for a System Type 1

The compensation technique for a **light tracking system** with a transfer function of Type 1 can be demonstrated with the aid of the following transfer function [21], [22]:

$$G_{P4}(s) = \frac{K'}{s(1+T_1s)(1+T_2s)} = \frac{50}{s(1+0.02s)(1+0.05s)} = \frac{500}{s^3 + 7s^2 + 100s} = \frac{K}{s^3 + 7s^2 + 100s} \quad (23)$$

Step 1: Evaluation of the Original Control System

The demonstration of the system's transient response is achieved by applying the following code:

```
>> Gp4=tf([0 500],[1 7 100 0])
Transfer function:
500
-----
s^3 + 7 s^2 + 100 s
>> Gfb4=feedback(Gp4,1)
>> step(Gfb4)
```

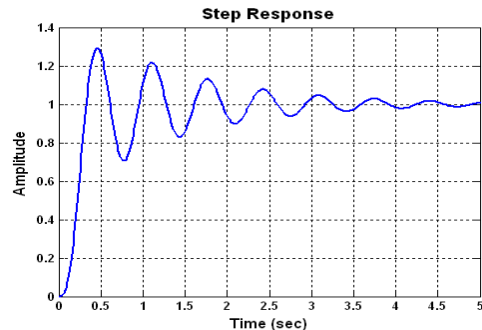


Figure 13: Transient Response of the Original System Type 1



Step 2: Determination of the Marginal System Gain

The critical marginal value of the unknown parameter K is achieved by means of applying the method of Advanced D-Partitioning. From equation (23), the characteristic equation of the system is obtained as:

$$G_p(s) = s^3 + 7s^2 + 100s + K \quad (24)$$

Reflecting on equation (24) the unknown value of K can be presented as follows:

$$K = -s^3 - 7s^2 - 100s \quad (25)$$

The D-partitioning curve in terms of the variable parameter K can be plotted in the complex plane within the frequency range $-\infty \leq \omega \leq +\infty$ facilitated by the following code and seen in Figure 14.

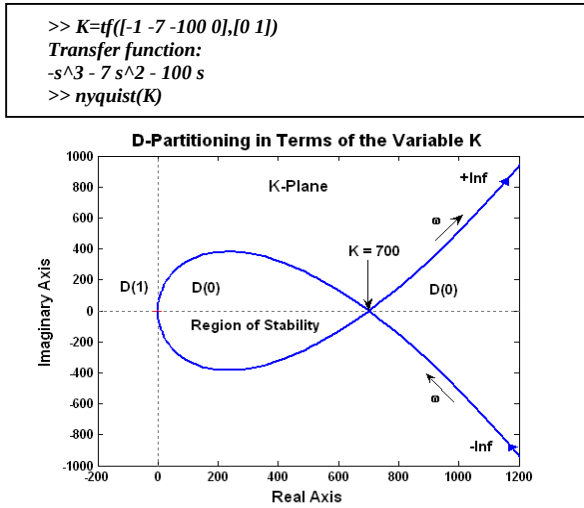


Figure 14: D-Partitioning in Terms of the Parameter K

The D-partitioning determines three regions on the K -plane: $D(0)$, $D(1)$ and $D(2)$. Only $D(0)$ is the region of stability, being the one, always on the left-hand side of the curve for a frequency variation from $-\infty$ to $+\infty$.

At $K = 700$, the system is marginal and the system's gain becomes $K' = 0.1K = 0.1 \times 700 = 70$.

Then equation (23) of the plant transfer function of Type 1 is modified to its marginal case as follows:

$$G_{p4}(s) = \frac{K'}{s(1+T_1s)(1+T_2s)} = \frac{70}{s(1+0.02s)(1+0.05s)} \quad (26)$$

Step 3: Optimization of ζ , t_s/t_m and PMO

For optimization of a marginal closed-loop system of Type 1, Rule 1 is used in the same manner, while Rule 2 is modified to Rule 5 and Rule 3 is ignored.

Rule 1: To optimize ζ , t_s/t_m and the PMO of a Type 1 marginal closed-loop system, a cascade multi-stage lead compensation with factors of $\alpha_{1,2,3,\dots}=10$ should be applied for a zero-pole cancellation. The number of the compensating stages N should be one less than the order of the open-loop system $N = n + m - 1$.



Rule 5: The pure integration or the most significant pole of the open-loop system should be left uncompensated. The existing system gain should be maintained by amplification equal to the product $\varepsilon\alpha_1\alpha_2\alpha_3\dots$, where the compensation amplification factor is $\varepsilon = (0.1 \text{ to } 1.27)$.

To keep $\zeta = 0.707$, when the ratio of the less significant to the most significant pole of the plant transfer function, $r = p_1/p_2$, varies from 50 to 1, the value of ε may vary from 0.1 to 1.27. The case $r = 2.5$, $\varepsilon = 1$ can be identified as shown in Figure 15 [23], [24], [25].

If the ratio is $r = 2.5$, as in the discussed case, then $\varepsilon = 1$. If $r < 2.5$, ε is within the limits $\varepsilon = (1 \text{ to } 1.27)$. If $r > 2.5$, then ε should be within the limits $\varepsilon = (0.1 \text{ to } 1)$.

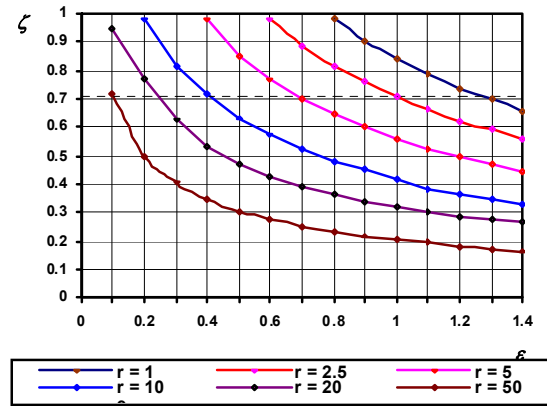


Figure 15: Relationship between the Damping Ratio ζ and the Factor ε for Different Poles Ratios $r = p_1/p_2$

For the plant with the transfer function represented by Equation (26), a two-stage lead compensation and attenuation is applied. The poles $p_1 = -50$ and $p_2 = -20$ are cancelled and amplification factors $\alpha_1 = \alpha_2 = 10$ and $\varepsilon = 1$ are used.

By applying Rule 1 and using a related sequence, the two-stage lead stage is presented by:

$$G'_c(s) = \frac{(1+\alpha_1 T_1 s)(1+\alpha_2 T_2 s)}{\alpha_1 \alpha_2 (1+T_1 s)(1+T_2 s)} = \frac{(1+0.02s)(1+0.05s)}{100(1+0.002s)(1+0.005s)} \quad (27)$$

Further from Rule 5:

$$G''_c(s) = \varepsilon \alpha_1 \alpha_2 = 1 \times 10 \times 10 = 100 \quad (28)$$

The transfer function of the complete compensator as a product of the three sections is:

$$G_c(s) = G'_c(s) \times G''_c(s) = \frac{100(1+0.02s)(1+0.05s)}{100(1+0.002s)(1+0.005s)} = \frac{(1+0.02s)(1+0.05s)}{(1+0.002s)(1+0.005s)} \quad (29)$$

By applying the full compensation, considering Equations (26) and (29), the continuous prototype of the transfer function of the open-loop compensated system becomes:



$$G(s) = G_c(s) \times G_{p4}(s) = \frac{70}{s(1 + 0.002s)(1 + 0.005s)} \quad (30)$$

Step 4: Conversion of the Continuous Prototype Compensated System into its Digital Equivalent.

The transient responses of the closed-loop digital control system before and after the compensation are shown in Figure 16 and are obtained by the code:

```
>> Gp=50000/(s+50)/(s+20)/(s+0)
>> Gpfbd = c2d(Gpfb,0.004,'tustin')
>> G=7000000/(s+500)/(s+200)/(s+0)
>> Gpfbd = c2d(Gpfb,0.002,'tustin')
>> step(Gpfbd,Gfbd)
```

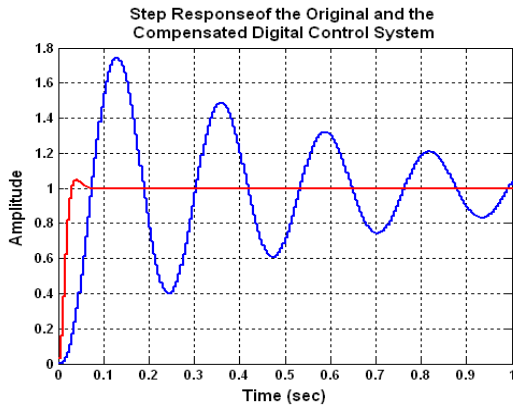


Figure 16: Comparison between the Transient Responses of the Original and Compensated Digital System Type I

The Bode diagrams, after applying the compensation technique, are achieved by the code as shown below and are presented in Figure 17.

```
>> Gpd=c2d(Gp,0.005,'tustin')
>> Gd=c2d(G,0.005,'tustin')
>> margin(Gd)
```

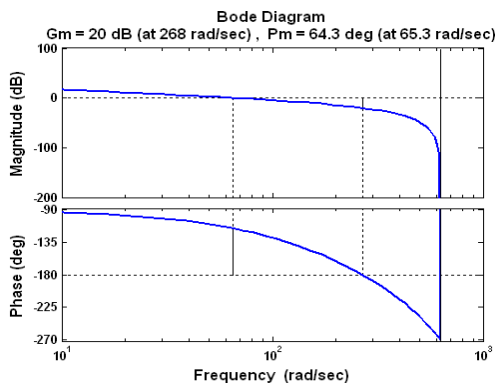


Figure 17: Bode Diagrams of the Digital Compensated System Proving Phase Margin PM = 64.3 and Damping $\zeta \approx 0.643$

Comparison between the results and the objectives for optimal performance is summarized in Table 3. It is seen from Figure 16, Figure 17 and Table 3 that the results after the digital compensation are better or a close match with the objectives.



Table 3
Objectives and Real Results for Case 3

Specifications	Objectives	Real Results	Consideration
ζ	= 0.707	= 0.643	Close
PMO	$\leq 4\%$	= 3.3%	Better
$t_s(1\%)/t_m$	≤ 1.49	= 1.4	Better
$e_{ss}(t)$	= 0%	= 0%	Matching

Step 5: Design of a Digital Compensation Controller based on a Microcontroller (Case 3).

If considering Equation (29), the transfer function of the modified compensation controller stage can be presented as:

$$G_c(s) = \frac{(1 + 0.02s)(1 + 0.05s)}{(1 + 0.002s)(1 + 0.005s)} = \frac{0.001s^2 + 0.07s + 1}{0.00001s^2 + 0.007s + 1} \quad (31)$$

Further, the digital equivalent of the compensator controller stage is realized by the code:

```
>> Gc=tf([0.001 0.07 1],[0.00001 0.007 1])
Transfer function:
0.001 s^2 + 0.07 s + 1
-----
1e-005 s^2 + 0.007 s + 1
>> Gcd = c2d(Gc,0.001,'tustin')
Transfer function:
75.29 z^2 - 145.4 z + 70.2
-----
z^2 - 1.418 z + 0.4909
Sampling time: 0.001
```

Then, the transfer function of the series controller stage in the discrete-time domain is as follows:

$$G_{cd}(z) = \frac{75.29z^2 - 145.4z + 70.2}{z^2 - 1.418z + 0.4909} = \frac{Y(z)}{X(z)} \quad (32)$$

Similarly to the other cases, a microcontroller can be connected in cascade with the original plant. To program the microcontroller, the transfer function of the series digital robust control stage is represented by the following difference equation [26], [27], [28]:

$$y(kT) = -1.418y[(k-1)T] + 0.4909y[(k-2)T] + 75.29x(kT) - 145.4x[(k-1)T] + 70.2x[(k-2)T] \quad (33)$$

Figure 18 demonstrates the analogue prototype of the compensated system.

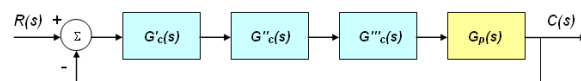


Figure 18: Block Diagram of the Analogue Prototype Compensated System



In Figure 19, the continuous model is converted into its digital equivalent by introducing the compensating microcontroller.

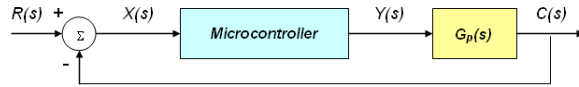


Figure 19: Block Diagram of the Digital Compensated System

4. Flowchart Diagram of the Analysis and Design Procedure Sequence

The sequence of the analysis and the design procedure are described by the following flowchart diagram [29].

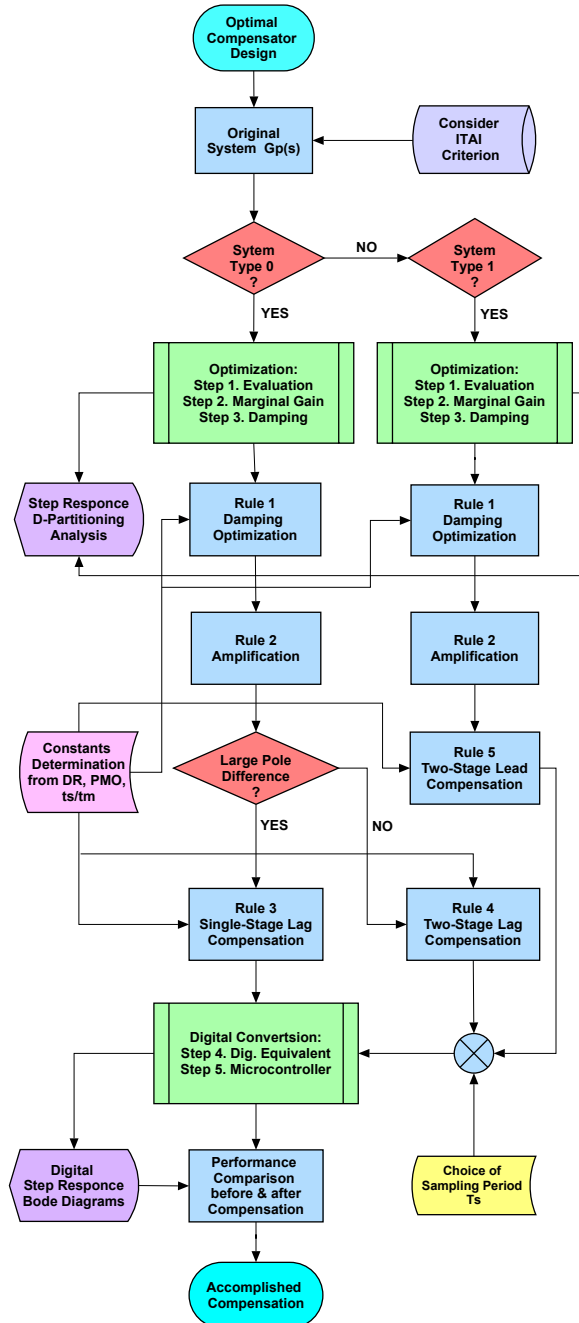


Figure 20: Flowchart Diagram of the Analysis and Compensator Design Procedure Sequence

This section describes the interactive procedure sequence [30], [31] for the analysis and design of the digital optimal compensation of the process control system.

The system identification is an important element of any process of analysis. The dynamics of the plant process control system is described by a differential equation and further its transfer function $G_P(s)$ can be derived. The ITAE criterion is considered for the design of the optimal compensator.

Further, the system is identified as Type 0 or Type 1. For any one of the cases evaluation of the system is performed in terms of transient response, marginal gain and relative damping. The design procedure sequence applying Rule 1 and Rule 2 follows in terms of relative damping optimization and proper amplification.

The constants α , γ , δ and ε are determined from their graphical relationships with the DR, PMO and t_s/t_m . These constants are further used for the implementation of Rule 1, Rule 3, Rule 4 and Rule 5.

The results are presented graphically after the initial system optimization, applying Step 1, Step 2 and Step 3 and displaying the original system transient response and gain margin original with the aid of the D-partitioning stability analysis.

If the system is Type 0, it is examined in terms of the difference between its dominant and its insignificant poles. If the difference is large, single stage lag compensation is applied. If the difference is insignificant, two-stage lag compensation is applied. For a system of Type 1, two-stage lead compensation is applied.

Finally, considering the Euler's approximation for choice of the proper sampling period T_s and applying the Bilinear Tustin Transform, digital conversion from the continuous-time presentation into the digital equivalent is performed by applying Step 4 and Step 5.

Taking into account the compensator's digital transfer function, its corresponding difference equation is derived based on which a microcontroller can be programmed to operate as a digital optimal compensator.

The results after the digital conversion are presented graphically in the digital time-domain in terms of the system's transient response and Bode diagram.

Comparison is completed for the system's performance before and after applying of the digital compensator and it is observed that an optimal compensation is accomplished.

5. Conclusion

The originality of the suggested technique of multi-stage compensation is based on the statement of a number of rules, which are applied in a predetermined sequence to some known theoretical procedures, like lead-lag compensation and zero-pole cancellation.

By analyzing, combining and applying these rules to control systems of different nature, some new ideas are developed, bringing exceptional final results of the system's performance.



The analogue compensation equipment consists of three major parts, facilitating the initial stage of the design.

The analogue lead compensator section eliminates all insignificant poles of the original plant transfer function and introduces new properly designed dominant poles, improving the transient response and especially the relative damping ratio of the system.

The amplifying section of the compensator reduces considerably the steady-state error.

The analogue lag compensator section eliminates the most significant pole of the original plant transfer function and improves further the transient response.

To upgrade further the compensator equipment, its analogue continuous transfer function is converted to its digital equivalent. This conversion is taking into account the Euler's approximation and is achieved by applying the Bilinear Tustin Transform, considered as one of the most accurate techniques.

The excellent final results, observed from the digital system performance, prove once again the perfect match between the continuous compensator and its digital equivalent, due to the application of the Euler's technique.

Difference equations are derived based on the digital equivalents of the compensators transfer functions. Microcontroller connected in cascade with the original plant can be programmed, based on the difference equation, reflecting the operation of the digital compensator.

The latest manufactured microcontrollers offer very high operating frequencies that can handle successfully the operation of each one of the discussed in this research control systems. For example the TMS320F2837xS is a powerful 32-bit floating-point microcontroller designed for advanced closed-loop control applications that provides 200 MHz of signal processing performance [32] and can be used for each one of the considered cases.

There is significant number of advantages associated with the digital control systems, compared with the continuous control systems. Due to this reason, the choice of compensation design strategy, resulting in the implementation of microcontrollers, is the better and preferable option and can be applied for any process control system.

6. References

- [1] Kuo B., Automatic Control Systems, 7th ed., New York: McGraw-Hill, pp.161-184, 2008.
- [2] Draper C.S., Principles of Optimising Control Systems, 2nd ed., American Society of Mechanical Engineers, USA, pp.86, 1991.
- [3] R.C. Dorf, Modern Control Systems, 3rd ed., New York, Addison-Wesley Publishing Company.
- [4] Shinnars J. S., Modern Control System Theory, 3rd ed., New York: McGraw-Hill, pp.190-203, 2004.
- [5] PID Controller Design, Retrieved March 28, 2017, <http://ctms.engin.umich.edu/CTMS/index.php?example=Introduction§ion=ControlPID>
- [6] Euler Method, In Wikipedia. Retrieved January 8, 2017, from http://en.wikipedia.org/wiki/Euler_method
- [7] Continuous-Discrete Conversion Methods. MathWorks R2014b Documentation. Retrieved January 18, 2017, <http://www.mathworks.com/help/control/ug/continuous-discrete-conversion-methods.html#bs78nig-1>
- [8] Phillips C.L., Digital Control System, Prentice-Hall International Inc., UK London, pp.125-403, 2005.
- [9] Yanev K.M., "Design and Analysis of a Robust Accurate Speed Control System by Applying a Digital Compensator", Journal of Automatic Control and System Engineering, USA, Volume 16, Issue 1, ISSN: 1687-4811, pp. 27-36, 2016.
- [10] Cruise-Control; Retrieved January 18, 2017, from <https://jagger.berkeley.edu/pack/me132/Section5.pdf>
- [11] Yanev K. M., "Application of the Method of D-Partitioning for Stability of Control Systems with Variable Parameters", BJT, Vol.16, Number 1, pp.51-58, 2007.
- [12] Yanev K. M., "Analysis of Systems with Variable Parameters and Robust Controller Design", Proceedings of the Sixth IASTED International Conference on Modelling, Simulation and Optimization, MSO 2006, ISBN 088986-618-X, pp.75-83, 2006.
- [13] Yanev K.M. Advanced D-Partitioning Stability Analysis in the 3-Dimensional Parameter Space, International Review of Automatic Control, ISSN: 1974-6059, Vol. 6, N. 3, pp. 236-240, 2013.
- [14] Yanev K.M., Obok Opok, "Improved Technique of Multi-Stage Compensation", First African Control Conference AFRICON 2003, Cape Town, South Africa, p.S1-S6, 2003.
- [15] Masupe S., Yanev K. M., "Design and D-Partitioning Analysis of Optimal Control System Compensation", Journal of International Review of Automatic Control, Vol. 4, N. 6, ISSN: 1974-6059, pp. 838-845, 2011.
- [16] Tustin with Frequency Prewarping. MathWorks R2014b Documentation. Retrieved January 17, 2017, <http://radio.feld.cvut.cz/matlab/toolbox/control/manipmod/ltiops20.html>.
- [17] Matched Z-transform method. In Wikipedia. Retrieved January 18, 2017, from http://en.wikipedia.org/wiki/Matched_Z-transform_method
- [18] Yanev K. M., Anderson G. O., Masupe S., Strategy for Analysis and Design of Digital Robust Control Systems, ICGST-ACSE Journal, Volume 12, Issue 1, pp. 37-44, 2012.



- [19] Abdala M. Alsharif M., DC Motor Drive System Cascade Control Strategy, Retrieved January 17, 2017, <https://www.slideshare.net/RishikeshBagwe/dc-motor-drive-system-cascade-control-strategy>
- [20] Behera L., Kar I., Intelligent Systems and Control Principles and Applications, Oxford University Press, pp. 20-83, 2009.
- [21] Yanev, K.M., Anderson G.O., Application of the D-partitioning for Analysis and Design of a Robust Photovoltaic Solar Tracker System, IJESCC, Volume 2, No. 1, pp. 43–54, 2011.
- [22] Yanev, K.M., Anderson G.O., Masupe S., D-partitioning Analysis and Design of a Robust Photovoltaic Light Tracker System, BIE 12th Annual Conference, 6001, ISBN: 97899912-0-731-5, 2011.
- [23] Yanev K.M., Advanced Interactive Tools for Analysis and Design of Nonlinear Robust Control Systems, International Review of Automatic Control, ISSN: 1974-6059, Vol. 6, N. 6, pp. 720-727, 2013.
- [24] Yanev K. M., “Strategy for Design of Optimal Control System Compensation”, International Journal of Energy Systems, Computers and Control, Vol. 1, No. 2, ISSN: 0976-6782, pp. 113–126, 2010.
- [25] Bhanot S., Process Control Principles and Applications, Oxford University Press, pp. 170-187, 2010.
- [26] Draper C.S., Principles of Optimizing Control Systems, 2nd ed., American Society of Mechanical Engineers, USA, pp.86, 1991.
- [27] Dorf R.C., Modern Control Systems, 3rd ed., New York, Addison-Wesley Publishing Company,
- [28] Yanev K.M., “Analysis of Systems with Variable Parameters and Robust Controller Design”, Sixth IASTED International Conference on Modelling, Simulation and Optimization, p. 75-83, 2006.
- [29] Click Charts Diagram & Flowchart Software Flexible Diagram Drawing and Creation, Retrieved March 28, 2017, from http://www.nchsoftware.com/chart/index.html?gclid=CJmBoOmkj9MCFTIW0wod9xsO_w
- [30] Lucidchart.com - Lucidchart Official Site Retrieved March 28, 2017, from <https://www.lucidchart.com/users/login?passwordOnly=1>
- [31] Flowchart Symbols Meaning Standard Flowchart Retrieved March 28, 2017, from <http://creately.com/diagram-type/objects/flowchart>
- [32] Texas Instruments; TMS320F2837xS Delfino™ Microcontrollers; Retrieved March 28, 2017, from <http://www.ti.com/lit/ds/symlink/tms320f28375s.pdf>

Biographies



Prof. Kamen Yanev has a M.Sc. in Control Systems Engineering. His Ph.D. is in the area of Systems with Variable Parameters and Robust Control Design. He started his academic career in 1974, following a number of academic promotions and being involved in considerable research, service and teaching at

different Commonwealth Universities around the world. Prof. Yanev worked in a number of outstanding universities in Europe and in Africa, in countries like Nigeria, Zimbabwe and Botswana.

Currently he works as Associate Professor in Control and Instrumentation Engineering at the Department of Electrical Engineering at University of Botswana.

His major research is in the field of Automatic Control and Systems Engineering as well as in the subjects of Electronics and Instrumentation. He has 114 publications in international journals and conference proceedings in the area of Control Systems, Electronics, Instrumentation and Power Engineering.

Most of his latest publications and current research interests are in the field of Electronics, Analysis of Control Systems with Variable Parameters, Nonlinear and Digital Control Systems, Robust Control Design and Optimal Process Control Systems.

Prof. Yanev is interacting for many years with Saint-Cyr French Military Academy, supervising Military Academy students doing their M.Sc. dissertations.

He is a member of the Institution of Electrical and Electronic Engineering (IEEE), a Gold member of the Academic Community of International Congress for Global Science and Technology (ICGST), a member of the Instrumentation, Systems and Automation Society (ISA) and a member of the Botswana Institute of Engineers (BIE).

Prof. Yanev is also a member of the Editorial Board of the International Journal of Energy Systems, Computers and Control (IJESCC), at International Science Press, a member of the Editorial Board of the Journal of Multidisciplinary Engineering Science and Technology (JMEST), a reviewer for ACTA Press and a reviewer for Advances in Science, Technology and Engineering Systems Journal (ASTESJ), USA.





Employing Vector Multiplication and Tilt Compensation Algorithm in Mimicking a Three Degree of Freedom Robotic Shoulder

J. Abliter, E. Agustin, M. Arroyo, A. Morante, J. Taduran, R. Tolentino

Electronics Engineering - Polytechnic University of the Philippines (Santa Rosa Campus)

Tagapo, Santa Rosa City, Laguna

johnmarkabliter@ymail.com, eugenenitsuga04@gmail.com, maerikaannearroyo@gmail.com,

mandreaalette@gmail.com, taduranjayson@ymail.com, kenmetara@yahoo.com

<https://www.pup.edu.ph/>

Abstract

The study aims to apply tri-axial accelerometers and magnetometer for acquiring angle in mimicking robotic shoulder using Vector Multiplication and Tilt Compensation Algorithm. The accelerometers and magnetometer are the main sensors where in the accelerometers are attached to the spine, right shoulder, upper arm and lower arm in slanting position of the user to acquire the pitch and roll movement. The magnetometer-accelerometer pair is attached on the top of the shoulder joint to acquire the yaw movement. The authors apply the concept of Vector Multiplication and Tilt Compensation Algorithm in the computation to obtain the angle of the human shoulder. The authors use C programming and LabVIEW Robotics software for extraction of the data to be used. Furthermore, the authors assert that the system is effective in acquiring the user's shoulder angle for mimicking robotic shoulder. Similarly, the authors analyze that the user's actual shoulder angle is close to the robotic shoulder prototype angle.

Keywords: *Accelerometers, Magnetometer, Shoulder Angle, Robotic Shoulder, Vector Multiplication, Tilt Compensation Algorithm*

Nomenclature

DOF	Degree-of-Freedom
IDE	Integrated Development Environment
NI VISA	National Instruments Virtual Instrument Software Architecture

1. Introduction

In the recent years, the development of interaction between robots and humans has become a key research topic in the field of robotics. Mimicking robotics is one of the important applications in the vast field of robotics, whereas the robots are being controlled by specific movements. Methods and techniques had been developed and different sensors have been used to capture human motion to manipulate the movements of the robot [1]. A lot of robots today use high cost integrated force torque sensors and cameras in order to detect human motion.

However, the most common and cheapest of these sensors are the accelerometers and magnetometers. Accelerometers can determine the position and orientation of an object. It is used to sense not only the gravity (tilt) but the sudden acceleration within a certain range of motion [2]. However, it had certain disadvantages like acquiring slow responses, being sensitive to acceleration forces due to movement and not knowing lateral orientation (yaw) because it only has a vertical reference (gravity) but not horizontal reference. On the other hand, magnetometers measure direction and strength of magnetic field along one dimension. It can be used to detect earth's magnetic field, which always points to north. This reference contains the horizontal component, from which it is possible to read off lateral orientation of the device, unlike accelerometers [3]. Combining these two sensors, it was possible to acquire angles that can be used to mimic human motion.

Various groups had attempted to create solutions in acquiring angles for mimicking human motion. Such works include a research work which attempted to create solutions to accurately translate human motions using Kinect Sensor by applying Vector Multiplication Approach such as Euclidean Distance, Cross Product, and Dot Product Approach [4]. Another work had determined the angular position of the elbow joint by attaching two tri-axial accelerometers to the upper arm and forearm of the user. The data obtained from the accelerometers were used to control the robotic elbow. [5] Another study combined kinematic models designed for control of robotic arms with state space methods to estimate angles of the human shoulder and elbow using gyroscope and accelerometer. The algorithm was tested on a 6-DOF (Degree of Freedom) industrial robotic arm wherein the shoulder joint was limited to only 2-DOF (pitch and yaw) movement [6]. The problem was that it only acquired the pitch and yaw angles of the human shoulder movement. If these acquired angles will be used to mimic the human shoulder movement, it will be inaccurate because the human shoulder requires 3-DOF which are the pitch, roll, and yaw movement.



In this paper, the main objective was to apply accelerometers to mimic the 2-DOF (pitch and roll) movement of the human shoulder using Vector Multiplication. The study can help the industry, in the manufacturing process. In this 3-DOF shoulder, the addition of 1-DOF also increases the robot's reach which means that the range of movement in each of its joints is considerably greater. This study can also help the medical field, in physical therapy and rehabilitation. The project is intended to aid in diagnosis of movement disorders. The magnetometer and accelerometer sensors are applied to mimic the remaining 1-DOF (yaw) movement of the human shoulder using Tilt Compensation Algorithm. The accelerometers are attached to the spine, upper arm, lower arm, and right shoulder of the user. The magnetometer-accelerometer pair is placed at the top of the shoulder joint. The robotic shoulder will move as the sensors measure the shoulder angles. The digital outputs of the sensors are read by the Arduino using LabVIEW Robotics which uses different mathematical approach to calculate the shoulder joint angles. This data is then feed to the servomotors by the Arduino using serial communication, for it to actuate the computed angles. Several tests are done to evaluate the proposed system. The results of the performed tests are presented and discussed.

2. Theory

The system is composed of a 3-DOF robotic shoulder equipped with Arduino Mega microcontroller, and actuated by three servo motors mounted at the robotic shoulder. Also, it is composed of a wearable device with the sensors. Another wearable device serves as measuring device comprised of potentiometer mounted on a 3-DOF mechanical joint that is used to know the actual shoulder angle produced by the user. Lastly, it is also composed of a computer running the program.

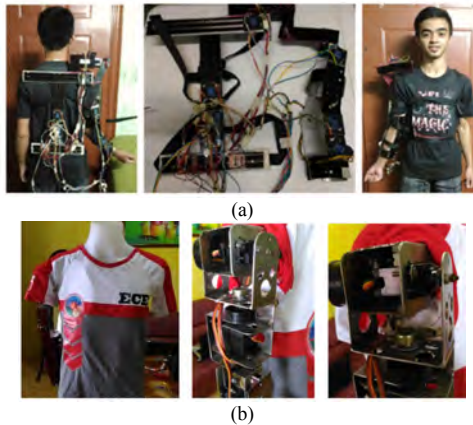


Figure 1. Wearable Device (a) and Robotic Shoulder (b)

The accelerometer and magnetometer in this study are the ADXL345 digital accelerometer and HMC5883L magnetometer. The communication between the sensors and the computer are coded in Arduino IDE (Integrated Development Environment). Figure 2 shows the explanation of how the system works. The authors used Arduino and LabVIEW software for the interfacing of program.

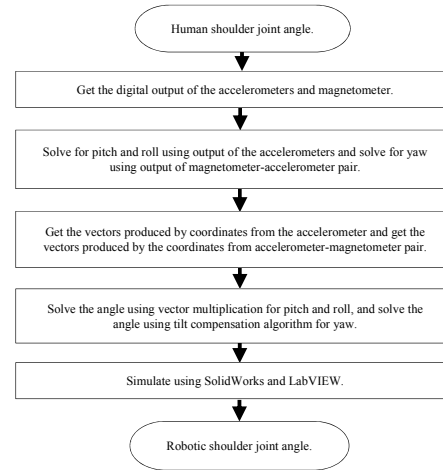


Figure 2. Conceptual Framework

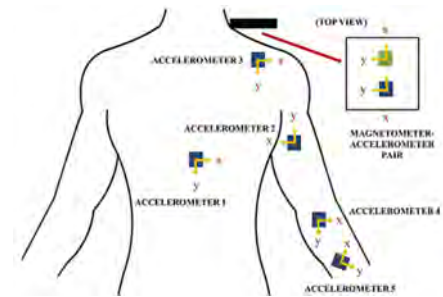


Figure 3. Selected points where the sensors are attached to the user

1. Steps in Acquiring Pitch and Roll Angle

The accelerometer has digital output in the range from -256 to +256 through 180° of tilt. The output in every axis (x-, y- and z-axis) is minimum if the axis is pointing downward, on the other hand, it has maximum value if it is pointing upward. The authors used these outputs to compute the acceleration due to gravity in different axes of the accelerometer which are attached to the user in the following selected points.

To the spine:

$$G'_{x1} = \frac{-DO_{x1}}{s} \quad G_{y1} = \frac{DO_{y1}}{s} \quad G_{z1} = \frac{DO_{z1}}{s} \quad (1)$$

Where:

DO_{x1} = digital output from the X-axis of the accelerometer
 DO_{y1} = digital output from the Y-axis of the accelerometer
 DO_{z1} = digital output from the Z-axis of the accelerometer
 S = the sensitivity of the accelerometer

G'_{x1} = acceleration from the X-axis of the accelerometer

G_{y1} = acceleration from the Y-axis of the accelerometer

G_{z1} = acceleration from the Z-axis of the accelerometer

To the upper arm:

$$G_{x2} = \frac{DO_{x2}}{s} \quad G'_{y2} = \frac{-DO_{y2}}{s} \quad G_{z2} = \frac{DO_{z2}}{s} \quad (2)$$

To the right shoulder:

$$G'_{x3} = \frac{-DO_{x3}}{s} \quad G_{y3} = \frac{DO_{y3}}{s} \quad G_{z3} = \frac{DO_{z3}}{s} \quad (3)$$

To the lower arm:

$$G'_{x4} = \frac{-DO_{x4}}{s} \quad G_{y4} = \frac{DO_{y4}}{s} \quad G_{z4} = \frac{DO_{z4}}{s} \quad (4)$$

To the lower arm in slanting position:



$$G'x_5 = \frac{-DOx_5}{s} \quad Gy_5 = \frac{DOy_5}{s} \quad Gz_5 = \frac{DOz_5}{s} \quad (5)$$

By using the value of acceleration due to gravity in different axes of the accelerometers, the authors compute the value of tilt angles which are static measurement of accelerometers by performing the definition of tangent function. These tilt angles are named as roll and pitch angles. To determine the value of pitch angles of the accelerometers, the proponents get the angle formed by the accelerometer with respect to vertical axis.

$$\phi_1 = \tan^{-1}\left(\frac{Gy_1}{G'x_1}\right) \quad (6) \quad \phi_2 = \tan^{-1}\left(\frac{G'y_2}{Gx_2}\right) \quad (7)$$

$$\phi_3 = \tan^{-1}\left(\frac{-Gy_3}{G'x_3}\right) \quad (8) \quad \phi_4 = \tan^{-1}\left(\frac{Gy_4}{G'x_4}\right) \quad (9)$$

$$\phi_5 = \tan^{-1}\left(\frac{Gy_5}{G'x_5}\right) \quad (10)$$

Where:

ϕ_1 = pitch angle of the accelerometer (spine)

ϕ_2 = pitch angle of the accelerometer (upper arm)

ϕ_3 = pitch angle of the accelerometer (right shoulder)

ϕ_4 = pitch angle of the accelerometer (lower arm)

ϕ_5 = pitch angle of the accelerometer (lower arm in slanting position)

To determine the value of and roll angles of the accelerometer the authors used the following formulas:

$$\theta_1 = \tan^{-1}\left(\frac{Gz_1}{\sqrt{(G'x_1)^2 + (Gy_1)^2}}\right) \quad (11) \quad \theta_2 = \tan^{-1}\left(\frac{Gz_2}{\sqrt{(Gx_2)^2 + (G'y_2)^2}}\right) \quad (12)$$

$$\theta_3 = \tan^{-1}\left(\frac{Gz_3}{\sqrt{(G'x_3)^2 + (-Gy_3)^2}}\right) \quad (13) \quad \theta_4 = \tan^{-1}\left(\frac{Gz_4}{\sqrt{(G'x_4)^2 + (Gy_4)^2}}\right) \quad (14)$$

$$\theta_5 = \tan^{-1}\left(\frac{Gz_5}{\sqrt{(G'x_5)^2 + (Gy_5)^2}}\right) \quad (15)$$

Where:

θ_1 = roll angle of the accelerometer (spine)

θ_2 = roll angle of the accelerometer (upper arm)

θ_3 = roll angle of the accelerometer (right shoulder)

θ_4 = roll angle of the accelerometer (lower arm)

θ_5 = roll angle of the accelerometer (lower arm in slanting position)

These values are used to find the vector (radius) components of the spherical coordinates. From the orientation and position of accelerometers attached to the user, the value of roll and pitch angles correspond to the value of azimuth and zenith angle in spherical coordinates respectively. To determine the value of unit vectors (which are Vectors A, B, C, d and E) of the accelerometers, the authors used the following formulas:

$$\vec{A} = [(\sin \phi_1)(\cos \theta_1)]x + [(\sin \phi_1)(\sin \theta_1)]y + [(\cos \phi_1)]z \quad (16)$$

$$\vec{B} = [(\sin \phi_2)(\cos \theta_2)]x + [(\sin \phi_2)(\sin \theta_2)]y + [(\cos \phi_2)]z \quad (17)$$

$$\vec{C} = [(\sin \phi_3)(\cos \theta_3)]x + [(\sin \phi_3)(\sin \theta_3)]y + [(\cos \phi_3)]z \quad (18)$$

$$\vec{D} = [(\sin \phi_4)(\cos \theta_4)]x + [(\sin \phi_4)(\sin \theta_4)]y + [(\cos \phi_4)]z \quad (19)$$

$$\vec{E} = [(\sin \phi_5)(\cos \theta_5)]x + [(\sin \phi_5)(\sin \theta_5)]y + [(\cos \phi_5)]z \quad (20)$$

For the pitch, the accelerometers attached to the spine and upper arm are used by connecting the selected joint coordinates, a vector will be formed. The magnitude of each of the vector produced by joint coordinates must be known to be able to calculate the pitch and roll angle. For the pitch, these vectors are named as Vector A and Vector

B for the imaginary lines from the spine to center shoulder and right shoulder to elbow, respectively.

The accelerometers attached to the right shoulder, upper arm and lower arm in slanting position are used for the roll movement. The vectors are named as Vector B, Vector C and Vector E for the imaginary lines from the derived point in the third accelerometer to right shoulder and derived point in the second accelerometer to right elbow, respectively.

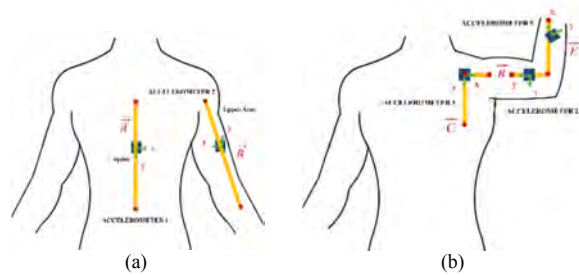


Figure 4. Construction of Vectors for (a) Pitch and (b) Roll

With the use of Vector Multiplication, the angle of the pitch and roll can be calculated. For the pitch movement, the authors apply the Dot Product and for the roll movement both Dot and Cross Product are applied.

For the pitch movement, Vector A and B uses Dot Product to acquire the angle.

$$\angle P = \cos^{-1}\left(\frac{\vec{A} \cdot \vec{B}}{|\vec{A}| |\vec{B}|}\right) \quad (21)$$

$$\angle P = \cos^{-1}\left(\frac{\left\{ \begin{aligned} &[(\sin \phi_1)(\cos \theta_1)(\sin \phi_2)(\cos \theta_2)] + \\ &[(\sin \phi_1)(\sin \theta_1)(\sin \phi_2)(\sin \theta_2)] + \\ &[(\cos \phi_1)(\cos \phi_2)] \end{aligned} \right\}}{\left\{ \begin{aligned} &\sqrt{(\sin \phi_1 \cos \theta_1)^2 + (\sin \phi_1 \sin \theta_1)^2 + (\cos \phi_1)^2} \\ &\sqrt{(\sin \phi_2 \cos \theta_2)^2 + (\sin \phi_2 \sin \theta_2)^2 + (\cos \phi_2)^2} \end{aligned} \right\}}\right) \quad (22)$$

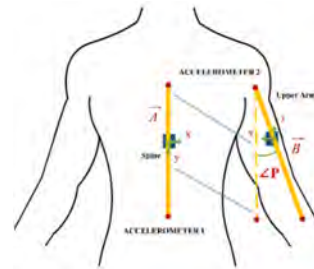


Figure 5. Calculation of Shoulder Angle for Pitch

For the roll movement, Vector F is calculated with the Cross Product of Vector C and B then, Vector G is calculated with the Cross Product of Vector E and the negative of Vector B. Then Dot Product is used in Vector F and G is used to acquire the roll angle. This is shown in the following formulas:

$$\vec{C} \times \vec{B} = \vec{F} = \begin{pmatrix} i & j & k \\ (\sin \phi_3)(\cos \theta_3) & (\sin \phi_3)(\sin \theta_3) & (\cos \phi_3) \\ (\sin \phi_2)(\cos \theta_2) & (\sin \phi_2)(\sin \theta_2) & (\cos \phi_2) \end{pmatrix} \quad (23)$$

$$\vec{F} = \left\{ \begin{aligned} &[(\sin \phi_3)(\sin \theta_3)(\cos \phi_2) - (\sin \phi_2)(\sin \theta_2)(\cos \phi_3)]i + \\ &[(\sin \phi_3)(\cos \theta_3)(\cos \phi_2) - (\sin \phi_2)(\cos \theta_2)(\cos \phi_3)]j + \\ &[(\sin \phi_3)(\cos \theta_3)(\sin \phi_2)(\sin \theta_2) - (\sin \phi_2)(\cos \theta_2)(\sin \phi_3)(\sin \theta_3)]k \end{aligned} \right\} \quad (24)$$



$$\begin{aligned} \vec{E} \times \vec{B} = \vec{G} &= \begin{pmatrix} i & j & k \\ (\sin \phi_2)(\cos \theta_2) & (\sin \phi_2)(\sin \theta_2) & (\cos \phi_2) \\ -(\sin \phi_2)(\cos \theta_2) & -(\sin \phi_2)(\sin \theta_2) & -(\cos \phi_2) \end{pmatrix} \\ \vec{G} &= \left\{ \begin{aligned} & [-(\sin \phi_2)(\sin \theta_2)(\cos \phi_2) + (\sin \phi_2)(\sin \theta_2)(\cos \phi_2)] i + \\ & [-(\sin \phi_2)(\cos \theta_2)(\cos \phi_2) + (\sin \phi_2)(\cos \theta_2)(\cos \phi_2)] j + \\ & [-(\sin \phi_2)(\cos \theta_2)(\sin \phi_2)(\sin \theta_2) + (\sin \phi_2)(\cos \theta_2)(\sin \phi_2)(\sin \theta_2)] k \end{aligned} \right\} \\ \angle R &= \cos^{-1} \left(\frac{\vec{F} \cdot \vec{G}}{|\vec{F}| |\vec{G}|} \right) \\ \angle R &= \cos^{-1} \left\{ \begin{aligned} & \left[\begin{aligned} & [(\sin \phi_2)(\sin \theta_2)(\cos \phi_2) - (\sin \phi_2)(\sin \theta_2)(\cos \phi_2)]^2 + \\ & [-(\sin \phi_2)(\sin \theta_2)(\cos \phi_2) + (\sin \phi_2)(\sin \theta_2)(\cos \phi_2)]^2 + \\ & [(\sin \phi_2)(\cos \theta_2)(\cos \phi_2) - (\sin \phi_2)(\cos \theta_2)(\cos \phi_2)]^2 + \\ & [-(\sin \phi_2)(\cos \theta_2)(\cos \phi_2) + (\sin \phi_2)(\cos \theta_2)(\cos \phi_2)]^2 \end{aligned} \right]^{1/2} \\ & \left[\begin{aligned} & [(\sin \phi_2)(\cos \theta_2)(\sin \phi_2)(\sin \theta_2) - (\sin \phi_2)(\cos \theta_2)(\sin \phi_2)(\sin \theta_2)]^2 + \\ & [-(\sin \phi_2)(\cos \theta_2)(\sin \phi_2)(\sin \theta_2) + (\sin \phi_2)(\cos \theta_2)(\sin \phi_2)(\sin \theta_2)]^2 \end{aligned} \right]^{1/2} \end{aligned} \right\} \\ \angle R &= \cos^{-1} \left\{ \begin{aligned} & \left[\begin{aligned} & [(\sin \phi_2)(\sin \theta_2)(\cos \phi_2) - (\sin \phi_2)(\sin \theta_2)(\cos \phi_2)]^2 + \\ & [(\sin \phi_2)(\cos \theta_2)(\cos \phi_2) - (\sin \phi_2)(\cos \theta_2)(\cos \phi_2)]^2 + \\ & [(\sin \phi_2)(\cos \theta_2)(\sin \phi_2)(\sin \theta_2) - (\sin \phi_2)(\cos \theta_2)(\sin \phi_2)(\sin \theta_2)]^2 \end{aligned} \right]^{1/2} \\ & \left[\begin{aligned} & [-(\sin \phi_2)(\sin \theta_2)(\cos \phi_2) + (\sin \phi_2)(\sin \theta_2)(\cos \phi_2)]^2 + \\ & [-(\sin \phi_2)(\cos \theta_2)(\cos \phi_2) + (\sin \phi_2)(\cos \theta_2)(\cos \phi_2)]^2 + \\ & [-(\sin \phi_2)(\cos \theta_2)(\sin \phi_2)(\sin \theta_2) + (\sin \phi_2)(\cos \theta_2)(\sin \phi_2)(\sin \theta_2)]^2 \end{aligned} \right]^{1/2} \end{aligned} \right\} \end{aligned}$$

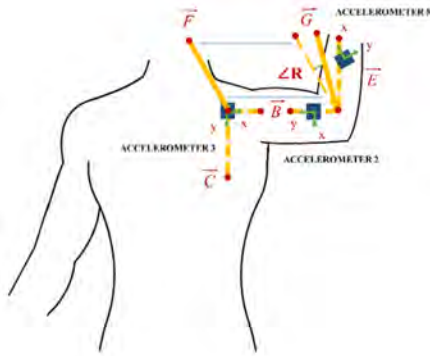


Figure 6. Calculation of Shoulder Angle for Roll

2. Steps in Acquiring Yaw Angle

The magnetometer-accelerometer pair is attached to the shoulder joint to mimic the yaw movement of the human shoulder. The magnetometer axes are the vectors acquired which is equal to the following:

$$MAG_{x-axis} = (DO_{MAGx-axis})(R_{MAG}) \quad (29)$$

Where:

$DO_{MAGx-axis}$ = digital output from the X-axis of the magnetometer in the shoulder joint

R_{MAG} = the resolution of the magnetometer

MAG_{x-axis} = vector from the X-axis of the magnetometer in the shoulder joint

$$MAG_{y-axis} = (DO_{MAGy-axis})(R_{MAG}) \quad (30)$$

$$MAG_{z-axis} = (DO_{MAGz-axis})(R_{MAG}) \quad (31)$$

After getting the magnetometer axes, the authors also acquired the normalized X-axis and Y-axis of the accelerometer attached to the shoulder joint of the user which is equal to the following:

$$N_{x-axis} = (DO_{ACCx-axis})(R_{ACC})(gf) \quad (32)$$

Where:

$DO_{ACCx-axis}$ = digital output from the X-axis of the accelerometer in the shoulder joint

R_{ACC} = the resolution of the accelerometer

gf = gravity factor

N_{x-axis} = normalized X-axis of the accelerometer in the shoulder joint

$$N_{y-axis} = (DO_{ACCy-axis})(R_{ACC})(gf) \quad (33)$$

By using the normalized axes, the authors computed the value of tilt angles which is the pitch and roll angle of the accelerometer.

$$\phi_6 = \sin^{-1}(N_{y-axis}) \quad (34)$$

Where:

N_{y-axis} = normalized Y-axis of the accelerometer in the shoulder joint

ϕ_6 = roll angle of the accelerometer (shoulder joint)

$$\theta_6 = \sin^{-1}(N_{x-axis}) \quad (35)$$

Where:

N_{x-axis} = normalized X-axis of the accelerometer in the shoulder joint

θ_6 = roll angle of the accelerometer (shoulder joint)

To determine the vectors to be used in the yaw movement of the shoulder, Tilt Compensation Algorithm is applied in order to acquire the X- and Y-axis of heading which are vectors in the magnetometer-accelerometer pair in the shoulder joint which is equal in the following:

$$X_h = (MAG_{x-axis})(\cos \phi_6) + (MAG_{z-axis})(\sin \phi_6) \quad (36)$$

Where:

MAG_{x-axis} = vector from the X-axis of the magnetometer in the shoulder joint

MAG_{z-axis} = vector from the Z-axis of the magnetometer in the shoulder joint

ϕ_6 = roll angle of the accelerometer (shoulder joint)

θ_6 = roll angle of the accelerometer (shoulder joint)

X_h = X-axis of heading

$$Y_h = \left\{ \begin{aligned} & (MAG_{x-axis})(\sin \phi_6)(\sin \theta_6) + (MAG_{y-axis})(\cos \theta_6) \\ & -(MAG_{z-axis})(\cos \phi_6)(\sin \theta_6) \end{aligned} \right\} \quad (37)$$

In acquiring the yaw movement properly and accurately, it is necessary to take into account the error factor: the magnetic declination. Magnetic declination is the angle on the horizontal plane between magnetic north (the direction the north end of a compass needle points, corresponding to the direction of the Earth's magnetic field lines) and true north (the direction along a meridian towards the geographic North Pole). This angle varies depending on the position on the Earth's surface, and changes over time (year). The value of the current magnetic declination can be found on the special magnetic maps, as well as navigation maps. The authors used a magnetic declination of 2°2' W (negative) based on the current location of the user. To convert the angle into radians, the authors used the following formula:

$$MagneticDeclinationAngle = \frac{\text{degree} + \frac{\text{minute}}{60}}{\frac{180}{\pi}} \quad (38)$$

To obtain the yaw movement, the proponents used the formula below:

$$\angle Y = \tan^{-1} \left(\frac{Y_h}{X_h} \right) \pm MagneticDeclinationAngle \quad (39)$$

To correct the yaw angle from 0° to 360°:



if $\angle Y < 0$
 $CorrectYawAngle = \angle Y + 2\pi$
 if $\angle Y > 0$
 $CorrectYawAngle = \angle Y - 2\pi$

To correct the rotation of the yaw angle, subtract the correct yaw angle from 360° :

$$\angle Y = 360^\circ - CorrectYawAngle \left(\frac{180^\circ}{\pi} \right) \quad (40)$$

3. Evaluation of the System

The authors devised a wearable device comprised of three potentiometers for each movement which is mounted on a 3-DOF mechanical joint. The proponents used the data coming from the accelerometers to acquire the shoulder angle by applying the mathematical methods discussed. The computed angle is read by the Arduino microcontroller. The authors also used potentiometer in the wearable sensor to compare data from the sensors. For the acquisition of prototype shoulder angle for pitch, roll and yaw, the authors devised a potentiometer-mounted robotic shoulder to translate its mechanical movement into analog signal.



Figure 7. Acquisition of Robotic Shoulder Angle

The shoulder angle values are sent through serial communication into a PC and then, inputted into LabVIEW's waveform chart feature which plotted the graph to monitor the response using NI VISA. In controlling the robotic arm, the authors used servo motor as an actuator. The servo motor is connected to Arduino microcontroller and is attached to the joint of robotic shoulder. The movement of servo motor is dependent to the value of computed angle. To know the critical value for a two-tailed test, the significance level (α) is set to 5%. Setting this significance value will create a confidence of 95% (obtained by $100\% - \alpha$), the area of the curve as the critical value is 0.975 (obtained by $1 - (\alpha/2)$). Knowing the area, the authors used the z-test table (area under the normal curve) and found the critical value 1.96. The authors will obtain the z value by using the z-test equation below:

$$z = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{(\sigma_1)^2}{n_1} + \frac{(\sigma_2)^2}{n_2}}} \quad (41)$$

Where:



z = z-test result

n_1 = no. of samples in the first group

n_2 = no. of samples in the second group

σ_1 = standard deviation of the first group

σ_2 = standard deviation of the second group

$\bar{X}_1$ = mean of the first group

$\bar{X}_2$ = mean of the second group

3. Results and Discussion

1. Acquisition of the Pitch and Roll Shoulder Angle

From the different angles made by the user, the authors obtained the data coming from the accelerometers. To analyze and interpret the computed angle, the authors gathered raw data through varying the movement of the human shoulder in random movements.

For Pitch: Table 1 and 2 shows the digital outputs in three different axes of accelerometers which are attached to the spine and upper arm, respectively. The accelerations due to gravity in every axis and the tilt angles are also shown. For Table 1, the vector is the orientation of the accelerometer in the spine with respect of the coronal plane of the user. For Table 2, the vector is the orientation of the accelerometer in the upper arm with respect to the arm of the user. From each angle the value of acceleration due to gravity in different axes and tilt angles are different from the other angles. It only shows that the user is moving from one angle to another.

Table 1. Data Obtained from the Accelerometer Attached to Spine

Shoulder Angle (Pitch)	Digital Output of Accelerometer			Acceleration Due To Gravity			Tilt Angles	
	DOx1	DOy1	DOz1	Gx1	Gy1	Gz1	$\Phi 1$	$\Theta 1$
20	-11	262	23	0.0430	1.0234	0.0898	87.5959	5.0125
30	-11	262	22	0.0430	1.0234	0.0859	87.5959	4.7956
40	-13	262	20	0.0508	1.0234	0.0781	87.1594	4.3599
50	-13	262	18	0.0508	1.0234	0.0703	87.1594	3.9254
60	-14	262	16	0.0547	1.0234	0.0625	86.9413	3.4897
70	-16	259	14	0.0625	1.0117	0.0547	86.4650	3.0882
80	-21	262	13	0.0820	1.0234	0.0508	85.4174	2.8315
90	-27	260	13	0.1055	1.0156	0.0508	84.0713	2.8471
100	-35	259	11	0.1367	1.0117	0.0430	82.3039	2.4101
110	-44	259	8	0.1719	1.0117	0.0313	80.3584	1.7442
120	-60	258	13	0.2344	1.0078	0.0508	76.9081	2.8097

Table 2. Data Obtained from the Accelerometer Attached to Upper Arm

Shoulder Angle (Pitch)	Digital Output of Accelerometer			Acceleration Due To Gravity			Tilt Angles	
	DOx2	DOy2	DOz2	Gx2	Gy2	Gz2	$\Phi 2$	$\Theta 2$
20	93	-227	3	0.3633	0.8867	0.0117	67.7215	0.7007
30	130	-208	4	0.5078	0.8125	0.0156	57.9946	0.9343
40	173	-189	6	0.6758	0.7383	0.0234	47.5308	1.3415
50	201	-152	18	0.7852	0.5938	0.0703	37.0973	4.0856
60	229	-115	21	0.8945	0.4492	0.0820	26.6650	4.6849
70	249	-75	20	0.9727	0.2930	0.0781	16.7626	4.3979
80	257	-25	18	1.0039	0.0977	0.0703	5.5560	3.9876
90	259	28	28	1.0117	-0.1094	0.1094	-6.1702	6.1347
100	248	81	43	0.9688	-0.3164	0.1680	-18.0877	9.3593
110	228	133	51	0.8906	-0.5195	0.1992	-30.2564	10.9356
120	193	184	46	0.7539	-0.7188	0.1797	-43.6325	9.7876

Table 3 shows the vector components of Vector A and B which are obtained from the accelerometers attached in the spine and the upper arm, the Scalar Product of the two vectors produced, the actual angle and computed shoulder angle, and their angle difference. The average of the angle difference obtained from this set of data was 0.2244° which is close to zero. It means that there is a small difference between the actual and computed shoulder angles. But still, the values of each computed pitch angle are close to the pitch shoulder angle of the user.



Table 3. Vector Components of the Accelerometers in the Spine and Upper Arm, the Scalar Product, the Comparison of Computed Shoulder Angle and Actual Shoulder Angle, and the Angle Difference for Pitch Movement

Shoulder Angle (Pitch)	A (Spine)			B (Upper Arm)			A · B	Computed Shoulder Angle	Angle Difference
	i	j	k	i	j	k			
20	0.9953	0.0873	0.0419	0.9253	0.0113	0.3791	0.9378	20.3108	0.3108
30	0.9956	0.0835	0.0419	0.8479	0.0138	0.5300	0.8676	29.8236	-0.1764
40	0.9959	0.0759	0.0496	0.7374	0.0173	0.6752	0.7692	39.7204	-0.2796
50	0.9964	0.0684	0.0496	0.6016	0.0430	0.7976	0.6420	50.0623	0.0623
60	0.9967	0.0608	0.0534	0.4473	0.0367	0.8936	0.4957	60.2827	0.2827
70	0.9966	0.0538	0.0617	0.2876	0.0221	0.9575	0.3468	69.7070	-0.2930
80	0.9956	0.0492	0.0799	0.0966	0.0067	0.9953	0.1760	79.8625	-0.1375
90	0.9934	0.0494	0.1033	-0.1069	-0.0115	0.9942	-0.0040	90.2314	0.2314
100	0.9901	0.0417	0.1339	-0.3063	-0.0505	0.9506	-0.1781	100.2600	0.2600
110	0.9854	0.0300	0.1675	-0.4947	-0.0956	0.8638	-0.3457	110.2249	0.2249
120	0.9728	0.0477	0.2265	-0.6800	-0.1173	0.7238	-0.5032	120.2100	0.2100
								Average Angle Difference	0.2244

For Roll: Table 4, 5 and 6 shows the digital outputs in three different axes of accelerometer attached to the right shoulder, lower arm in slanting position and upper arm, respectively. The accelerations due to gravity in every axis and the tilt angles are also shown.

Table 4. Data Obtained from the Accelerometer Attached to Right Shoulder

Shoulder Angle	Accelerometer Attached to the Right Shoulder							
	Digital Output of Accelerometer			Acceleration Due To Gravity			Tilt Angles	
	DOx3	DOy3	DOz3	Gx3	Gy3	Gz3	Φ3	θ3
0	-11	278	-1	0.0430	1.0859	-0.0039	87.7341	-0.2059
10	0	277	-7	0.0000	1.0820	-0.0273	90.0000	-1.4476
20	2	278	-10	-0.0078	1.0859	-0.0391	90.4122	-2.0601
30	-6	276	-12	0.0234	1.0781	-0.0469	88.7546	-2.4890
40	-8	277	-12	0.0313	1.0820	-0.0469	88.3457	-2.4795
50	-7	277	-14	0.0273	1.0820	-0.0547	88.5524	-2.8924
60	-6	278	-17	0.0234	1.0859	-0.0664	88.7636	-3.4985
70	-1	279	-13	0.0039	1.0898	-0.0508	89.7946	-2.6677
80	-5	276	-17	0.0195	1.0781	-0.0664	88.9621	-3.5241
90	-5	277	-20	0.0195	1.0820	-0.0781	88.9659	-4.1290
100	-6	275	-21	0.0234	1.0742	-0.0820	88.7501	-4.3658

Table 5. Data Obtained from the Accelerometer Attached to Lower Arm in Slanting Position

Shoulder Angle	Accelerometer Attached to the Lower Arm in Slanting Position							
	Digital Output of Accelerometer			Acceleration Due To Gravity			Tilt Angles	
	DOx5	DOy5	DOz5	Gx5	Gy5	Gz5	Φ5	θ5
0	206	-157	1	0.8047	0.6133	0.0039	142.6877	0.2212
10	214	-153	-52	0.8359	0.5977	0.2031	144.4370	11.1814
20	207	-144	-100	0.8086	0.5625	0.3906	145.1755	21.6319
30	184	-134	-142	0.7188	0.5234	0.5547	143.9356	31.9576
40	163	-114	-180	0.6367	0.4453	0.7031	145.0316	42.1430
50	135	-87	-212	0.5273	0.3398	0.8281	147.2005	52.8535
60	96	-64	-238	0.3750	0.2500	0.9297	146.3099	64.1368
70	62	-51	-253	0.2422	0.1992	0.9883	140.5599	72.3950
80	16	-24	-258	0.0625	0.0938	1.0078	123.6901	83.6208
90	-25	1	-257	0.0977	0.0039	1.0039	2.2906	84.4395
100	-34	8	-259	0.1328	0.0313	1.0117	13.2405	82.3195

For Table 4, this vector is the orientation of the accelerometer in the right shoulder with respect to the coronal plane of the user. For Table 5, this vector is the orientation of the accelerometer in the lower arm in slanting position with respect to the arm of the user. And for Table 6, this vector is the orientation of the accelerometer in the upper arm with respect to the arm of

the user. From each angle the value of acceleration due to gravity in different axes and tilt angles are different from the other angles. It only shows that the user is moving from one angle to another.

Table 6. Data Obtained from the Accelerometer Attached to Upper Arm

Shoulder Angle	Accelerometer Attached to the Upper Arm							
	Digital Output of Accelerometer			Acceleration Due To Gravity			Tilt Angles	
	DOx2	DOy2	DOz2	Gx2	Gy2	Gz2	Φ2	θ2
0	255	-13	35	0.9961	0.0508	0.1367	2.9184	7.8053
10	255	-13	-15	0.9961	0.0508	-0.0586	2.9184	-3.3621
20	251	-14	-64	0.9805	0.0547	-0.2500	3.1925	-14.2832
30	227	-14	-98	0.8867	0.0547	-0.3828	3.5292	-23.3112
40	204	-2	-139	0.7969	0.0078	-0.5430	0.5617	-34.2682
50	172	3	-181	0.6719	-0.0117	-0.7070	-0.9992	-46.4561
60	128	4	-202	0.5000	-0.0156	-0.7891	-1.7899	-57.6264
70	95	-1	-220	0.3711	0.0039	-0.8594	0.6031	-66.6433
80	45	-1	-231	0.1758	0.0039	-0.9023	1.2730	-78.9739
90	1	-3	-232	0.0039	0.0117	-0.9063	71.5651	-89.2191
100	-14	-3	-233	-0.0547	0.0117	-0.9102	167.9052	-86.4836

Table 7 shows the vector components of Vector B, Vector C and Vector E which are obtained from the accelerometers attached in the upper arm, lower arm in slanting position and the right shoulder.

Table 7. Vector Components of the Accelerometers in the Upper Arm, Lower Arm in Slanting Position and Right Shoulder for Roll Movement

Shoulder Angle	B (Upper Arm)			C (Right Shoulder)			E (Lower Arm)		
	i	j	k	i	j	k	i	j	k
0	0.0504	0.0069	0.9987	0.9992	-0.0036	0.0395	-0.6062	-0.0023	-0.7953
10	0.0508	-0.0030	0.9987	0.9997	-0.0253	0.0000	-0.5706	0.1128	-0.8135
20	0.0540	-0.0137	0.9984	0.9993	-0.0359	-0.0072	-0.5308	0.2105	-0.8209
30	0.0565	-0.0244	0.9981	0.9988	-0.0434	0.0217	-0.4995	0.3116	-0.8084
40	0.0681	-0.0055	1.0000	0.9986	-0.0432	0.0289	-0.4250	0.3846	-0.8195
50	-0.0120	0.0126	0.9998	0.9984	-0.0504	0.0253	-0.3271	0.4318	-0.8406
60	-0.0167	0.0264	0.9995	0.9979	-0.0610	0.0216	-0.2420	0.4991	-0.8321
70	0.0042	-0.0097	0.9999	0.9989	-0.0465	0.0036	-0.1921	0.6055	-0.7723
80	0.0042	-0.0218	0.9998	0.9979	-0.0615	0.0181	-0.0924	0.8269	-0.5547
90	0.0129	-0.9486	0.3162	0.9972	-0.0720	0.0180	0.0039	-0.0398	0.9992
100	0.0129	-0.2091	-0.9778	0.9969	-0.0761	0.0218	0.0306	-0.2270	0.9734

Table 8 shows the vector components of the cross product of Vector C and B which is equal to Vector F and the vector components of the cross product of Vector E and Vector -B which is equal to Vector G. Table 8 also shows the magnitudes of Vector F and Vector G and its Dot Product.

Table 8. Vector Components from the Cross Product and Dot Product Using Vectors B, C and E, for Roll Movement

Shoulder Angle	F = C x B			G = E x -B			F	G	F · G
	i	j	k	i	j	k			
0	-0.0039	-0.9959	0.0071	-0.0032	-0.5652	0.0041	0.9960	0.5653	0.5630
10	-0.0252	-0.9984	-0.0017	-0.1102	-0.5285	0.0040	0.9987	0.5399	0.5392
20	-0.0360	-0.9982	-0.0118	-0.1989	-0.4857	0.0041	0.9989	0.5249	0.5243
30	-0.0428	-0.9957	-0.0219	-0.2913	-0.4528	0.0054	0.9969	0.5385	0.5368
40	-0.0431	-0.9984	-0.0052	-0.3800	-0.4183	0.0008	0.9993	0.5651	0.5647
50	-0.0508	-0.9986	0.0120	-0.4423	-0.3372	-0.0011	0.9999	0.5562	0.5561
60	-0.0615	-0.9978	0.0253	-0.5208	-0.2558	-0.0020	1.0000	0.5803	0.5803
70	-0.0465	-0.9988	-0.0095	-0.5980	-0.1889	0.0007	1.0000	0.6271	0.6271
80	-0.0610	-0.9976	-0.0215	-0.8146	-0.0901	0.0015	0.9997	0.8196	0.8193
90	-0.0056	-0.3151	-0.9450	-0.9353	-0.0032	0.9962	0.9962	0.9353	0.9318
100	0.0790	0.9750	-0.2075	-0.4255	-0.0424	0.0035	1.0000	0.4276	0.4276

Table 9 shows the comparison of the actual angle and the computed shoulder angle and the angle difference between them. The average of the angle difference obtained is 0.2496° which is close to zero. It means that there is a small difference between the actual and computed shoulder angles. But still, the values of each computed pitch angle are close to the pitch shoulder angle of the user.



Table 9. Comparison of the Actual Shoulder Angle, Computed Shoulder Angle and Angle Difference for Roll Movement

Shoulder Angle (Roll)	Actual Shoulder Angle	Computed Shoulder Angle	Angle Difference
0	0	0.0986	0.0986
10	10	10.3451	0.3451
20	20	20.2360	0.2360
30	30	30.3458	0.3458
40	40	39.7852	-0.2148
50	50	49.7807	-0.2193
60	60	60.3323	0.3323
70	70	69.8052	-0.1948
80	80	80.1935	0.1935
90	90	89.6323	-0.3677
100	100	100.1977	0.1977
Average Angle Difference			0.2496

2. Acquisition of the Yaw Shoulder Angle

From the different angles made by the user, the authors obtained the data coming from the magnetometer-accelerometer pair. The table for digital outputs of magnetometer, normalized axes, tilt angles, x and y axis of heading, the computed and actual shoulder angle, and the total difference in each angle.

Table 10. Data Obtained from the Magnetometer-Accelerometer Pair Attached to the Shoulder Joint

Shoulder Angle (Yaw)	MAGX-axis	MAGY-axis	MAGZ-axis	nX-axis	nY-axis	Φ_6	Θ_6	Xh	Yh
0	379.04	11.12	-139.84	-0.02	0	1.1460	0	376.1674	11.1200
10	384.48	-55.53	-140.76	0.02	-0.01	-1.1460	-0.5730	387.2183	-56.8576
20	382.50	-126.90	-140.76	0.01	0	-0.5730	0	383.8885	-126.9000
30	363.40	-195.96	-140.76	0.01	0	-0.5730	0	364.7894	-195.9600
40	330.44	-263.96	-142.60	0.03	0	-1.7191	0	334.5693	-263.9600
50	294.60	-322.92	-140.76	0.04	-0.02	-2.2924	-1.1460	290.0026	-325.4407
60	226.48	-365.24	-139.84	0.02	-0.03	-1.1460	-1.7191	229.2315	-369.1341
70	154.72	-397.44	-138.00	0.03	-0.02	-1.7191	-1.1460	158.7904	-400.0264
80	78.28	-412.16	-132.48	0.06	-0.02	-3.4398	-1.1460	86.0878	-414.6285
90	10.12	-411.24	-131.56	0.02	0.06	-1.1460	3.4398	12.7492	-402.6192

Table 10 shows the data obtained from the Magnetometer-Accelerometer Pair attached to the shoulder joint. The produced value of vector in magnetometer which are the MAGX-axis, MAGY-axis and the MAGZ-axis, the normalized values of the axes of the accelerometer which are the N_X -axis and N_Y -axis, the tilt angles, and the axes of headings which are X_h and Y_h of the magnetometer-accelerometer pair are shown in this table.

Table 11. Comparison of the Actual Shoulder Angle, Computed Shoulder Angle and Angle Difference for Yaw Movement

Shoulder Angle (Yaw)	Actual Shoulder Angle	Computed Shoulder Angle	Angle Difference
0	0	0.3401	0.3401
10	10	10.3867	0.3867
20	20	20.3254	0.3254
30	30	30.2774	0.2774
40	40	40.3052	0.3052
50	50	50.3289	0.3289
60	60	60.1931	0.1931
70	70	70.3827	0.3827
80	80	80.3039	0.3039
90	90	90.2196	0.2196
Average Angle Difference			0.3063

Table 11 shows the comparison of the actual angle and the computed shoulder angle and the angle difference between them. The average of the angle difference obtained is 0.3063° which is close to zero. It means that there is a small difference between the actual and computed shoulder angles. But still, the values of each computed pitch angle are close to the pitch shoulder angle of the user.

3. Evaluation of the Significant Difference between Robotic Shoulder Angle Human Shoulder Angle

The user performs random movements to test the capability and the accuracy of the prototype to mimic its pitch, roll and yaw movements in different angles.

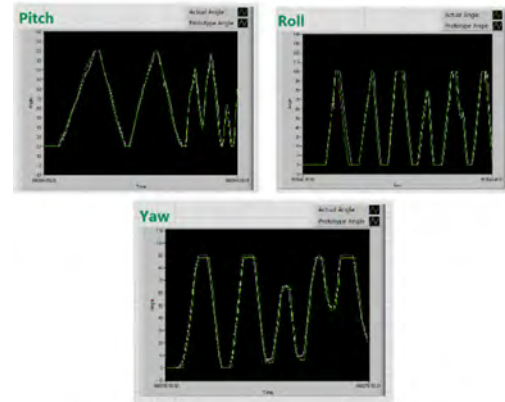


Figure 8. Lab VIEW Panel Showing the Graph of Actual and Prototype Angle for Pitch, Roll and Yaw Movement

The graph shows the actual and the prototype angle for the shoulder movements. The actual and the prototype angle for pitch, roll and yaw are always almost the same to each other. It is observed in the graph that the white line has more jitters than the green line, it simply indicates that the robotic shoulder has more noise than the actual shoulder. The noise and jitters are at minimum therefore, the response of the prototype to the user is realistic. Overall, it shows that the system has a good response in different shoulder movement. Below are the tables showing the pitch, roll and yaw movement for every angle and the results of z-test from the actual angle and robotic shoulder angle for pitch, roll and yaw.

Table 12, 13 and 14 shows the number of sample (n_1 and n_2), the mean of the sample ($\bar{x}_1$ and $\bar{x}_2$), the standard deviation of the sample (σ_1 and σ_2) and the z-test result. The results of the z-test for the pitch, roll and yaw shoulder movement are -0.2641, -0.3569 and -0.418, respectively. The results are close to zero. It means that the angular difference of actual angle and prototype angle is at minimum. It also means that the response is reliable. Therefore, the z-test result is in the acceptance region. Overall, there is no significant difference between the actual angle and the robotic shoulder angle for pitch, roll and yaw shoulder movement.

Table 12. Z-Test results for every Angle for Pitch Movement

Actual angle			Prototype angle			Result of Z-test	Comment
n_1	$\bar{x}_1$	σ_1	n_2	$\bar{x}_2$	σ_2		
600	66.9317	32.2157	600	67.4167	31.3973	-0.2641	Null Hypothesis is Accepted

Table 13. Z-Test results for every Angle for Roll Movement

Actual angle			Prototype angle			Result of Z-test	Comment
n_1	$\bar{x}_1$	σ_1	n_2	$\bar{x}_2$	σ_2		
425	44.8424	34.0517	425	45.68	34.375	-0.3569	Null Hypothesis is Accepted

Table 14 Z-Test results for every Angle for Yaw Movement

Actual angle			Prototype angle			Result of Z-test	Comment
n_1	$\bar{x}_1$	σ_1	n_2	$\bar{x}_2$	σ_2		
410	46.6073	32.9803	410	47.5659	32.6761	-0.4181	Null Hypothesis is Accepted



4. Conclusion

The authors came with this findings based on the gathered data. The values of the acceleration due to gravity in different axis, pitch and roll angles obtained from the accelerometers varies in different angle, but it arrives with the angle close to the user's shoulder angle movement for pitch and roll movement. The average angle difference of the pitch, roll and yaw movement are close to zero which means that the method is effective for mimicking the shoulder's pitch and roll angle. This suggests that the system, the Vector Multiplication, Tilt Compensation Algorithm method proposed are effective in acquiring the value of user's pitch, roll and yaw shoulder angle. However, there is still slight angle difference between the actual shoulder angles and the computed shoulder angles due to unstableness of the accelerometer's tilt angles that easily affects the computed shoulder angles which can be seen in the inconsistency of the increasing and decreasing values of the tilt angles produced by each accelerometer. In terms of statistical data gathered by the authors, all of the z-test results for shoulder angles at different movement are within the range of -1.96 and +1.96, which means that the values are all accepted. Therefore, the prototype can effectively mimic the user's shoulder movements but it is also observed that there are some fluctuations of signal in the graphical response of the system. It is because of the noise and jitters caused by the system.

5. Recommendation

For future works, the authors recommend to use other methods that will minimize the effect of the unstableness of the accelerometer's tilt angles on the computed shoulder angles. It also suggests to use filtering methods on the system to lessen the jitter or noise that affects the response of the robotic shoulder. Thus, the future researchers may improve the response of the system.

6. References

- [1] Ameri, Naghshineh and MazdakZereshki, "Human Motion Capture Using Tri-Axial Accelerometers", Institute of Electrical and Electronics Engineers Xplore (IEEE Xplore), 2009.
- [2] Moreira, Pires and Pedro Neto, "Accelerometer Based Control of an Industrial Robotic Arm", Institute of Electrical and Electronics Engineers Xplore (IEEE Xplore), 2009.
- [3] Madgwick, Sebastian, "Automated Calibration of Accelerometers, Magnetometers and Gyroscopes", Institute of Electrical and Electronics Engineers Xplore (IEEE Xplore), 2010.
- [4] Emano, Mapalad, Siron, Ortega and Tolentino, "Vector Multiplication Approach for Point of View Variations in a Mimicking Robotic Shoulder Using Microsoft Kinect", International Journal of Inventive Engineering and Sciences (IJIES), Vol. 4, 2016.
- [5] Alcira, Balla, Baris, Bilon, Burgos and Tolentino, "Scalar Product in Acquiring Angle for Mimicking Robotic Elbow using Two Tri-Axial Accelerometers", International Journal of Innovative Science and Modern Engineering (IJISME), Vol.4, 2016.

- [6] El-Gohary, Mahmoud Ahmed, "Human Joint Angle Estimation with Inertial Sensors and Validation with a Robot Arm", Portland State University Dissertations and Theses, 2015.

Biographies



John Mark M. Abliter is currently studying the degree in Bachelor of Science in Electronics and Communications Engineering from the Polytechnic University of the Philippines Santa Rosa Campus, Laguna in 2017.



Eugene V. Agustin is currently studying the degree in Bachelor of Science in Electronics and Communications Engineering from the Polytechnic University of the Philippines Santa Rosa Campus, Laguna in 2017.



Ma. Erika Anne P. Arroyo is currently studying the degree in Bachelor of Science in Electronics and Communications Engineering from the Polytechnic University of the Philippines Santa Rosa Campus, Laguna in 2017.



Andrea Alette R. Morante is currently studying the degree in Bachelor of Science in Electronics and Communications Engineering from the Polytechnic University of the Philippines Santa Rosa Campus, Laguna in 2017.



Jayson C. Taturan is currently studying the degree in Bachelor of Science in Electronics and Communications Engineering from the Polytechnic University of the Philippines Santa Rosa Campus, Laguna in 2017.



Roselito E. Tolentino is a registered Electronics Engineer and IECEP-Member. He is a graduate of Bachelor of Science in Electronics and Communication Engineering at Adamson University in 2004. He finished his Master of Science in Electronics Engineering Major in Control System at Mapua Institute of Technology. He currently takes up Doctor of Philosophy in Electronics Engineering at the same Institute. He is currently working as a part time instructor at PUP - Santa Rosa Campus and DLSU-Dasmariñas.





A Hybrid Controller for Inflight Stability and Maneuverability of an Unmanned Aerial Vehicle in Indoor Terrains

C. Todd¹, H. Koujan², S. Fasciani³

¹Faculty of Computer Science, Hawaii Pacific University, 1164 Bishop Street, Honolulu, Hawaii, 96813, USA

^{2,3}Dept. Information Science and Engineering, University of Wollongong in Dubai, Knowledge Village, Dubai, UAE

¹cghourani@hpu.edu, ²hk963@uowmail.edu.au, ³stefanofasciani@uowdubai.ac.ae

¹<http://www.hpu.edu>, ^{2,3}<http://www.uowdubai.ac.ae>

Abstract

A hybrid controller is designed and tested in simulation-based experiments, overcoming several research challenges associated with stability and control of UAV flight in a GPS-denied environment, achieving faster more accurate autonomous indoor flight. There exists a need for design and modeling of a robust, innovative flight controller that offer more stable, autonomous flight in challenging indoor environments. Drone drifts caused by air turbulence pose as a significant challenge for autonomous aerial systems that are GPS-denied. A hybrid flight controller, that utilizes PID, PID² and FLC techniques in addition to KF, EKF and Complimentary Error Minimization algorithms, has been developed in a simulation software and validated. Major contributions of the work include: multiple switchable flight modes at the Position and Tracking Controller (PTC) level, and multiple quadcopter flying configuration at the Motor Mixer level. A novel drift correction mechanism utilizes the drone states and an EKF estimator generates a compensation signal with integration of the PTC. FLC adds an extra layer of control by determining the drone's flight mode and flying configuration for the optimal flight output and stability performance. Contributions of this work may further offer increase in accuracy for connected processes enabling UAV flight, including SLAM, path planning, collision detection and avoidance and, subsequently, overall flight performance.¹

Keywords: Autonomous UAV, Flight Controller, Drift Correction, Hybrid Controller, Fuzzy Logic and PID Controller.

1. Introduction

Unmanned Aerial Vehicles (UAVs) are those capable of flight controlled by an onboard computer and/or remotely from the ground. The UAV can be designed to support numerous tasks such as surveillance, scientific research,

construction and military operations [1]. UAVs can be classified based on their lift mechanisms such as fixed wings, rotorcrafts or Vertical Take Off and Landing (VTOL) [2]. A quadcopter is a multirotor aircraft built using four thrust sources, generally fixed angle propellers driven by DC motors [2]. Compared to other types of aerial vehicles, quadcopters offer several advantages: the single rigid body with four fixed angle propellers provides a simple mechanical structure, while other aircrafts such as helicopters use one rotor and require complex mechanical components to change the rotor orientation while providing thrust to maneuver [3]. A quadcopter can execute movements with higher accuracy than helicopters, essential when operating in indoor environments [3]. Quadcopters can fly at low speed, whereas fixed-wing aircrafts do not present such flexibility in aerodynamics and require high air pressure for flight capability that occurs only at a high speed [3]. However, quadcopters present major drawbacks in terms of power efficiency, due to continuous, rapid acceleration of the four motors and flight stability. Effective control strategies may be introduced for stability improvements [4].

In this work, a novel approach to controller design is proposed toward enhanced UAV (Quadcopter) flight stability and maneuverability, overcoming several research challenges within this field. A Quadcopter possesses structural characteristics that change with its dynamic model and each model poses unique design challenges. Appropriate modeling of aerodynamics and dexterous control are the main concerns in this field. To establish autonomous flight accuracy of pose estimation, obstacle location and classification are key issues [5][6]. Indoor environments raise special concerns in terms of nature of failures and safety issues. Current UAV flight controller techniques can handle sparse and large indoor areas, but embedded and/or moving objects of various type and shape still place significant restrictions on indoor drone flight capability. This context requires sophisticated

¹ This study has been implemented on a Simulink platform in Matlab.



and advanced control algorithms to circumnavigate obstacles, for path planning and localization. Sophisticated control algorithms may be designed to enable a UAV to circumnavigate indoor embedded obstacles, providing dexterity in control strategy. Indoor spaces deny the UAV's accessibility to signals provided by the Global Positioning System (GPS), and as such, alternative solutions, for drone localization and/or as input for control, are needed. A control system that communicates with path planning and collision avoidance is required to output correct flight path decisions for the drone's controller and also to facilitate UAV inflight stability, which is subject several disturbances in indoor contexts [1], [4], [6], [7], [8]. Air turbulence reflected from objects close to the drone, including when taking off or landing, causes lateral drifts that affect the accuracy of the autonomous functions [1], [7]. Light-weight UAVs introduce additional challenges in dynamic modeling because the assumption made for larger drones, such as structural rigidity and symmetry, are no longer valid and non-linear parameters in the mathematical model of the Quadcopter frame should be introduced [6]. Simultaneous changes in the drone's motors speeds add another level of complexity that may determine in instability [8].

For UAV autonomous control, research challenges are of a technical nature and are application-specific, such as relating to drift, instability and inaccuracies due to inappropriate fine-tuning of the flight controller. This work overcomes associated research challenges by designing a hybrid flight controller system offering stability and high accuracy in indoor autonomous flight. A control algorithm that achieves better flight stability for indoor autonomous quadrotors as compared to existing solutions [9], [10], [11], [12] is designed, implemented and validated using a hybrid model of Proportional, Integral, Derivative (PID) and Fuzzy Logic Control (FLC), in Matlab Simulink. UAV design exhibits improved performance offering faster response time and accuracy of flight control, in comparison to existing techniques [1], [2]. This system overcomes disturbances of drift in which better results of mapping, localization, collision detection and avoidance may be obtained, within GPS-denied environments.

Section (2) focuses on the hybrid controller design and implementation in Simulink, in Matlab. Section (3) presents the results of the study. Section (4) provides a discussion of the research findings and Section (5) emphasizes major conclusions and recommendations of the work.

2. Controller Design and Implementation

In the applied method, it is assumed that the drone has full operability of all propellers and that the battery provides supply power for the duration of the flight. Environmental issues such as lateral drifts during drone-obstacle air turbulence interference are considered for controller design. A standard quadcopter is considered, as in [1], [7], [13], with rigid body construction and four thrust sources (motors) placed on each wing, equi-spaced from the frame center, as in Figure 1. Propeller rotation direction plays an important role in achieving quadrotor basic stability [7]; each two opposing fans, separated by 180° , should rotate in the same direction, with direction opposite to that of the

other fan pair. For example, if propellers 1 and 3 are rotating clockwise, propellers 2 and 4 should rotate anti-clockwise [13]. Quadrotor motion is described with its tilt on three axes of rotation w_x , w_y and w_z [13]. The UAV rotation around the w_z (Yaw) axis results in turning to the left or right, and it can be achieved by reducing the speed of one of the counter rotating propeller pairs [1]. Rotation of the quadrotor about the w_y (Pitch) axis is performed by rising the angular speed of motor 4 and decreasing the speed of motor 2 [1]. Rotation about the w_x (Roll) axis is controlled by the same operation as for Pitch but with fans 1 and 3 [1]. The UAV's take-off and landing mechanism simultaneously increases or decreases the speed of all propellers [2]. Basic maneuvers of a quadrotor assist in understanding stability requirements and flight control of a UAV of this type. The controller design proposed controls six degrees of freedom including yaw, pitch, roll, x-transition, y-transition and attitude of the four motors positioned on the quadrotor. This is achieved through three layers of control.

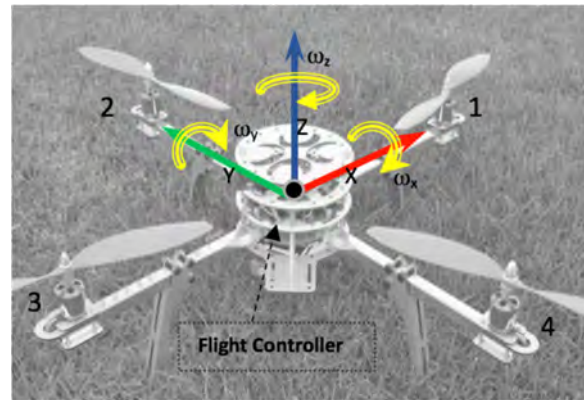


Figure 1. Quadcopter: six Degree of Freedom, rigid frame with four equi-spaced thrust sources.

PID control is combined with FLC to provide a hybrid solution for greater quadrotor control and stability during indoor autonomous flight. Several PID controllers are embedded enabling multiple inputs for more complex functions demanded by the drone [9]; a technique that offers a higher control accuracy of each degree of freedom of the quadcopter [8], [10]. The FLC requires that control functions are written using linguistics for all possible control scenarios, forming the control data sets [5]. Compared to other methods, FLC do not require an accurate input signal: smaller errors and overshoots of system output, and faster time to reach steady state produce better results in terms of final stability [2], [14]. The Backstepping control technique is not considered in this design due to the high computational complexity, which is not suitable for UAVs controlled by onboard low power computational platforms. Backstepping is a recursive complementary tool that has been used with other control techniques to achieve better stability than singular control techniques [5], [10], [15]. A major challenge in designing and implementing the Backstepping controller is the greater complexity and accuracy in terms of the mathematical and dynamic models required to fully integrate them in a quadrotor. The



Backstepping controller is substituted by the Kalman Filter (KF) and the Extended Kalman Filter (EKF) that enable estimating the result of a process with a high degree of confidence. The EKF algorithm is designed to estimate the state of a non-linear systems affected by unpredictable noises. EKF uses less computational resources and it requires less information about the estimated system than a Backstepper controller [16]. The aforementioned techniques have been implemented to minimize the measurement errors and to estimate the drift of the quadcopter as part of the drift correction mechanism.

A hybrid controller that exploits the advantages of all the three controllers: PID, FLC and EKF has been designed, developed and tested in this work. PID controllers are designed to control independently pitch, roll, yaw and altitude; one controller is dedicated to each degree of freedom. The FLC is designed to then achieve further flight stability; it continuously adjusts the flight mode and the motor control mixer configuration. Further, the FLC operates as the main controller that determines the movement output of the UAV based on two parameters; the input signal and the estimated disturbance existing in the drone's environment. An overview of the proposed system is illustrated in Figure 2.

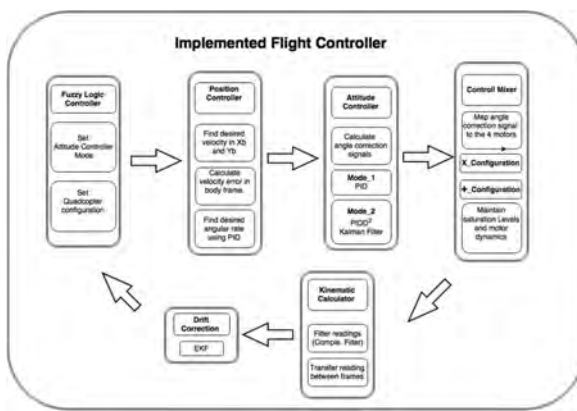


Figure 2. Configuration of the FLC, PID and EKF controllers within the hybrid controller design.

The drift correction algorithm (Figure 3) designed for this flight controller, utilizes an EKF to estimate the real pitch and roll of the drone away from the control processes. Then, the estimated angles are fed back to the flight controller as an error signal after subtracting them from the input command signals. This error, the difference between the two signals, represents the drift of the quadcopter that will be corrected automatically by the flight controller as it computes the other needed corrections.

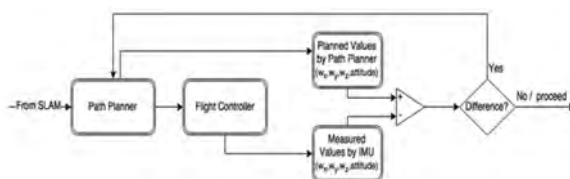


Figure 3. Flowchart of the algorithm for Drift Correction.

Modeling of Drone Control: The controller utilizes multiple techniques in control and filtering including PID, Proportional, Integral, Derivative (Second Order) PID², FLC, KF, complementary filtering and Extended Kalman Filter (EKF). The flight controller is divided into five components: Position and Tracking Controllers, a Motor Mixer, a Drift Correction mechanism and the FLC. Novel techniques are included in each component of the UAV flight control and stabilization: the Position Controller compensates for drift correction by application of the EKF in addition to applying existing techniques including basic manipulation of the desired x-, y- and z- coordinate, which represent the desired rotational rate as output. The Tracking Controller receives the output from the Position Controller and together with the proposed control strategies, determines the correction signal required and adjusts the quadcopter's attitude. Existing techniques implement only a single control methodology for the Tracking Controller for all flight conditions. In this work the Tracking Controller has two flight modes with dynamic switching capability for mode selection, to overcome induced stability and fast-response issues. The motor control mixer then maps the angle correction signals determined by the Tracking Controller into the required throttle for each of the four motors on the UAV frame. The quadcopter can fly in two different kinematic configurations: X configuration, in which the drone's X-Y axes of movements are 45 degrees apart between each of the four motors and the Plus (+) configuration, in which the drone's body frame is the same as its inertial one. In this work, there is an option to switch between configurations based on the desired output which is a novel approach.

For autonomous drone flight, drifts are key concerns. To address these, the EKF has been designed and implemented to estimate environmental drifts. This drift correction mechanism estimates pitch and roll angles in isolation from control process noise and interference; these estimates are then compared to the planned angles for drift calculation. The compensation commands are then generated by the Position Controller. PID and associated derivatives are implemented in the Flight Controller for individual motor control. To improve system stability, the FLC is implemented to determine the optimal setting required to drive the most stable output: given the multiple operating flight modes and configurations of the Flight Controller, the FLC operates to drive mode and configuration selection for greatest stability.

Mechanical Modeling: Mathematical modeling of quadcopter aerodynamics enables calculation of the required practical parameters for controller implementation. Two conventional coordinate systems are for the quadcopters are the Plus (+) configuration, with the x- and y- axes as the arms supporting the two consecutive motors (90° arm separation). According to the Plus convention the first motor must rotate anti-clockwise and the second motor clockwise. In the Ex (X) configuration, the x-axis is aligned by +45° from the arm of the first motor towards the second; that is, the Plus configuration rotated by +45°. Changes in configuration will cause differences in modelling and alignment of motors during this process. In this work, the both



configurations are enabled with switching between each mode.

Assuming rigidity of the quadcopter's structure and propellers, that the UAV structure is symmetrical, and the center of gravity and fixed body frame coincide, the moment of inertia of the quadcopter mass may be defined by the following matrix (Equation 1):

$$J^b = \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \quad \text{Equation 1}$$

By applying approximations [11], [12], the momentum of the quadcopter's structure, actuator mass and arm mass are determined by Equations 2, 3 and 4.

$$J_{xx} = J_{x,Motor} + J_{x,Hub\ Body} + J_{x,Arm} \quad \text{Equation 2}$$

$$\begin{aligned} = & \left[2 \left[\frac{1}{4} m_M r_M^2 + \frac{1}{3} m_M h_M^2 \right] \right. \\ & + 2 \left[\frac{1}{4} m_M r_M^2 + \frac{1}{3} m_M h_M^2 \right. \\ & \left. \left. + m_M d_M^2 \right] \right] \\ & + \left[\left[\frac{1}{4} m_H r^2 + \frac{1}{12} m_H h_H^2 \right] \right] \\ & + \left[2 \left[\frac{1}{2} m_A r_A^2 \right] \right. \\ & + 2 \left[\frac{1}{4} m_A r^2 + \frac{1}{3} m_A L_A^2 \right. \\ & \left. \left. + m_A d_A^2 \right] \right] \end{aligned}$$

$$J_{yy} = J_{y,Motor} + J_{y,Hub\ Body} + J_{y,Arm} \quad \text{Equation 3}$$

$$\begin{aligned} = & \left[2 \left[\frac{1}{4} m_M r_M^2 + \frac{1}{3} m_M h_M^2 \right] \right. \\ & + 2 \left[\frac{1}{4} m_M r_M^2 + \frac{1}{3} m_M h_M^2 \right. \\ & \left. \left. + m_M d_m^2 \right] \right] \\ & + \left[\left[\frac{1}{4} m_H r^2 + \frac{1}{12} m_H h_H^2 \right] \right] \end{aligned}$$

$$\begin{aligned} & + \left[2 \left[\frac{1}{2} m_A r_A^2 \right] \right. \\ & + 2 \left[\frac{1}{4} m_A r^2 + \frac{1}{3} m_A L_A^2 \right. \\ & \left. \left. + m_A d_A^2 \right] \right] \\ J_{zz} = J_{z,Motor} + \\ & J_{z,Hub\ Body} + J_{z,Arm} \end{aligned} \quad \text{Equation 4}$$

$$\begin{aligned} = & 4 \left[\left[\frac{1}{2} m_M r_M^2 + m_M d_m^2 \right] \right] + \left[\frac{1}{2} m_H r_H^2 \right] + \\ & \left[4 \left[\frac{1}{4} m_A r_A^2 + \frac{1}{3} m_A L_A^2 + m_A d_A^2 \right] \right] \end{aligned}$$

where m_x is the mass of the sub-notation x which could be referring to the motor (M), drone's arm (A) or drone's hub body (H), d_x is the radius from what is referred to in the sub-notation x to the Centre of Mass, h_x is the height of sub-notation x from its base, L_x is the total length from the base to sub-notation x , and r_x is the radius of the structure referred to in the sub-notation x .

The main forces and moments that affect the drone are from gravity or due to the thrust and torque initiated by the four rotors [3]. The thrust generated by the propellers are conventionally forces that are perpendicular to the x - y plane of the body frame in the positive direction of the z axes [12]. Drone maneuverability is constrained by motor thrust, which is the key driving force for its movements [11]. The generated thrust, T , by an actuator on the quadcopter is described by Equation 5.

$$T = C_T \rho A_r r^2 \omega^2 \quad \text{Equation 5}$$

where C_T is the thrust coefficient provided by the manufacturer of the rotor or measured manually, ρ is the air density, A_r is the propeller's rotation cross-sectional area, r is the radius of the rotor and ω is its angular velocity. Simplifying Equation 5, Equation 6 represents the lumped thrust coefficient, C_T .

$$T = c_T \omega^2 \quad \text{Equation 6}$$

The total thrust provided by all four propellers is represented by Equation 7. Varying the thrusts of motors 2 and 4 results in rotation of the drone about the x -axis. This rotation is coined the rolling torque, L , as represented by Equation 8.

$$T_{tot} = T_{motor1} + T_{motor2} + T_{motor3} + T_{motor4} \quad \text{Equation 7}$$



$$L = l(T_{motor4} - T_{motor2})$$

Equation 8

where l is the arm length of the drone. The pitch torque, M , is the drone's rotation around y-axis and results due to a change in the thrust of motors 1 and 3. Pitch torque is calculated by application of Equation 9.

$$M = l(T_{motor1} - T_{motor3})$$

Equation 9

The yawing torque, N , is the drone's rotation around the z-axis. It is the resultant of the difference in the developed torques between all four motor rotations (counter-clockwise and clockwise) and is expressed as Equation 10.

$$N = l[(T_{motor2} + T_{motor4}) - (T_{motor1} + T_{motor3})]$$

Equation 10

The torque generated by each motor, Q , can be expressed using the lumped torque coefficient, and is represented by Equation 11.

$$Q_{motor_x} = c_Q \omega_{motor_x}^2$$

Equation 11

Practical tests enable determination of the parameters in Equations 1 to 11. All inertial forces and torques of the quadcopter may then be grouped into one, inertial, matrix (Equation 12).

$$\begin{bmatrix} \Sigma T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} c_T & c_T & c_T & c_T \\ 0 & d_+ c_T & 0 & -d_+ c_T \\ -d_+ c_T & 0 & d_+ c_T & 0 \\ -c_Q & c_Q & -c_Q & c_Q \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}$$

Equation 12

where d is just another convention notation for the aforementioned notation l .

For modeling of the mechanical system, all parameters and forces that affect the aerodynamics of the UAV may be captured in one matrix. The resulting matrix, $M_{A,T}^b$, (Equation 13) accounts for all of the aerodynamic, gyroscopic, and thrust moments created by the rotors'

systems on the quadcopter for the Plus configuration. The lift force is represented by Equation 14. Additional effects such as blade flapping and frame aerodynamic drag not included in the model.

$$M_{A,T}^b =$$

$$\begin{bmatrix} d_+ c_T \omega_2^2 - d_+ c_T \omega_4^2 + J_m Q \left(\frac{\pi}{30} \right) (\omega_1 - \omega_2 + \omega_3 - \omega_4) \\ -d_+ c_T \omega_1^2 + d_+ c_T \omega_3^2 + J_m P \left(\frac{\pi}{30} \right) (-\omega_1 + \omega_2 - \omega_3 + \omega_4) \\ -c_Q \omega_1^2 + c_Q \omega_2^2 - c_Q \omega_3^2 + c_Q \omega_4^2 \end{bmatrix}$$

Equation 13

PID Controller Model: The quadcopter PID flight controller consists of two sub-systems: the first determines the desired quadcopter rotational rate driving position control, and the second measures the actual quadcopter rotational speed for calculation of the required motor output to achieve the desired rotational movement, as shown in Figure 4. The Inertial Measurement Unit (IMU) attains actual rotational position information pertaining to the rotational speed of the drone in addition to the angle of deviation from any of the drone axes. This actual angle measurement is compared with the desired angle of the drone, which represent the input of the first PID controller. The difference of desired rotational rate to meet the requirements is sent to the second PID controller and compared with the actual rotational rate to provide motors with a throttle control signal (voltage).

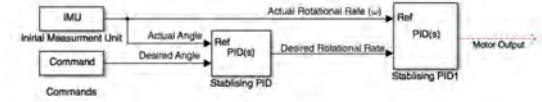


Figure 4. Diagram of the two sub-systems for PID control of motor angular rotation.

Dynamic alternation is achieved in-flight between different stability and structural modes. The Tracking Controller processes the quadcopter commanded desired position in two stages: first it finds the feed forward error of the quadcopter's position to determine the desired rotational velocities, and then it finds the required correction to be performed by the drone for its rotational states and altitude. The Motor Mixer Controller translates the required corrections into throttle commands that are distributed to each of the four motors and are dependent on the quadcopter configuration. Both Plus and Ex configurations integrated in the system, requiring a switching strategy to provide maximum drone maneuverability. The State Finder manipulates all parameters, state equations and forces acting on the drone for current use by the system.

The Tracking Controller determines the correction needed for pitch, roll and yaw of the drone, in addition to the altitude, for velocity control; by comparing actual and desired position and attitude readings throughout several layers of PID control. An improved controller is achieved over existing solutions [9], [10], [11], [12] with novelty in utilization of two modes (configurations) in combination with FLC for tuning and a PID² controller with a KF to



filter out noise. This approach further provides a more accurate measurement of the angle of the quadcopter. Variation of the Tracking Controller from existing designs lie in the dual mode switching capabilities dependent on the flight environment, in addition to the KF implementation to calculate more exactly the error signal of the feed-forward PIDD². The KF effectively eliminates noise due to measurement and control to increase the accuracy of the correction signal made by the controller. After receiving the desired x and y coordinates, altitude and yaw, the Tracking Controller calculates the desired pitch and roll angles from these inputs, applying Euler's Equations to compute the transformation of the drone's position and velocities between its inertial- and body-frames. First the error is determined for the x and y coordinates of the body-frame. This error is then mapped to a desired velocity in the same frame, which is transferred to the desired rotational rate. For calculation of error and its mapping to desired rotational rates, the yaw (Psi angle, Ψ) is needed since a quadcopter can maneuver towards a point with different Yaw angles. The x and y components of the yaw are scaled by the difference between the current coordinates and intended ones to find the desired velocity in the x and y axes of the body frame, as in Equation 15 and 16 [3]. Figure 5 shows the Simulink design of the Tracking Controller.

$$\begin{aligned} U_{Xb,Desired} &= X_{error} * \cos(\psi) + Y_{error} * \sin(\psi) \\ &= (X_{cmd} - X_{actual}) * \cos(\psi) + (Y_{cmd} - Y_{actual}) * \sin(\psi) \quad \text{Equation 15} \end{aligned}$$

$$\begin{aligned} U_{Yb,Desired} &= Y_{error} * \cos(\psi) - X_{error} * \sin(\psi) \\ &= (Y_{cmd} - Y_{actual}) * \cos(\psi) - (X_{cmd} - X_{actual}) * \sin(\psi) \quad \text{Equation 16} \end{aligned}$$

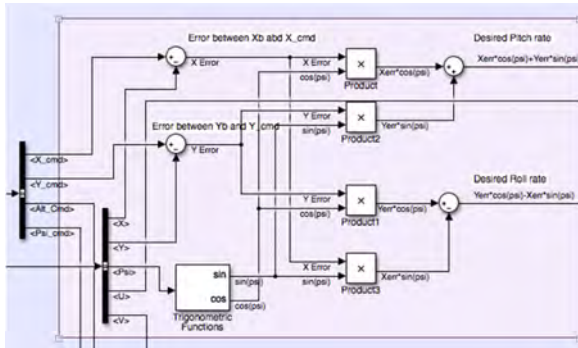


Figure 5. Simulink design for modeling of the mathematical manipulation needed to determine the desired velocity correction in x and y axes of the body-frame using the signals coming from the commands and quadcopter state manipulator.

Actual pitch and roll rates are controlled by PID controllers that use the desired velocities and actual velocities, U and V , measured by the IMU. The feed-forward PID technique is applied since it offers faster response than a feedback PID system for control of electric motors [15], [17]. The feed-forward subsystem calculates the required adjustment needed by each rotational drone

axis; this is sent as a command signal to the next subsystem to direct drone movement by the quantified angle about the respective axis. For calculation of the angle of required rotation Θ , the difference between the pitch rate and the linear velocity is calculated along the x-axis of the inertial-frame. This provides the velocity error, which is sent to a feed-forward PD controller to compute the required Θ , limited within the physical characteristics of the drone and input to the next section of the controller. For calculation of the angle of rotation about the y-axis, the roll rate, the desired roll rate comprises the positive input while the negative input is the linear velocity in the y-axis of the inertial-frame. A PD controller enables the required compensation in terms of roll to be found. The sign inversion corrects the calculated error due to mathematical operations.

The pitch θ , roll ϕ , yaw Ψ and altitude correction signals are computed in the following subsystem of the controller using feed-forward PID controllers. This process can be done in one of two modes: the Acrobatic mode for fast, sharp maneuvers or the Extra Stable mode for additional stability. In Acrobatic mode, first the error is determined and is then fed into the proportional and integral components of the controller (Figure 6). This error is the difference between desired and actual the drone's rotational angle measurements. The derivative component comprises a scaling factor since rotational rates of the drone may be determined through analysis of the IMU readings and application of Euler's Equations; substituted as the derivative of the corresponding angle. All four required correction signals are generated via identical controllers with the exception of altitude because this has a gravity offset input to achieve hovering at a specific height. Further, the altitude controller finds the derivative of the linear velocity on the z-axis to calculate the vertical acceleration of the drone which assists in its stabilization at the defined gravity offset.

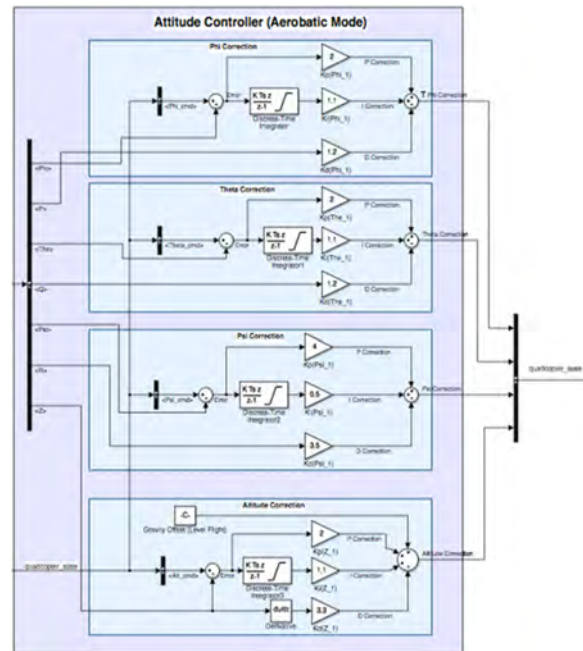


Figure 6. Simulink implementation of the Aerobic flight controller mode.



The Extra Stable mode for flight control detailed in Figure 7 enables higher flight stability with a tradeoff in slower response time due to the KF applied to the input signal for noise reduction, providing accurate estimates of pitch (θ) and roll (ϕ) angles. The derivative component in the PID controller projects the future error of the plant signal and also enables the controlled system to stabilize about a targeted point. This type of controller is a PIDD² due to the second derivative component and prove useful in flight systems such as this in achieving with greater accuracy and stability, the desired attitude or position. Angle correction calculations in this mode are performed using a similar PIDD² structure, with altitude correction requiring an additional component for consideration of initial conditions, such as if the drone had a specified starting altitude (gravity offset).

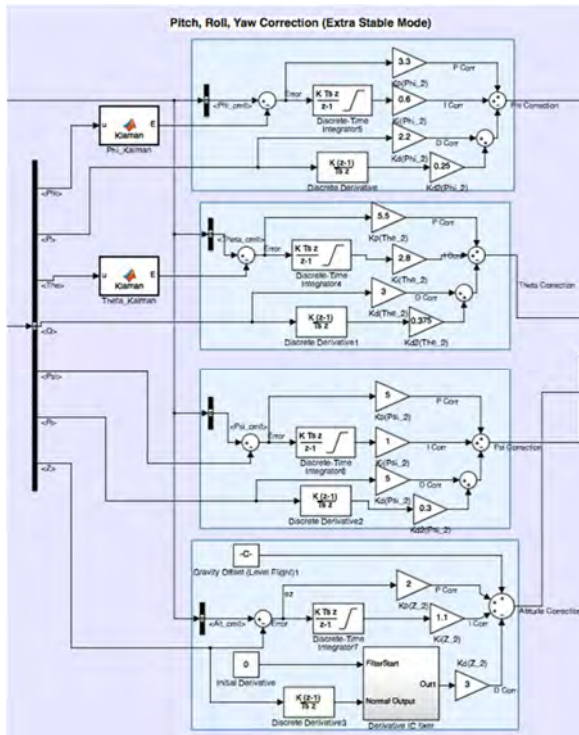


Figure 7. Simulink implementation of the Extra Stable flight controller mode.

The KF is implemented as a Matlab function box, which is a customized Simulink block. The state model of this KF, A , includes the angle and it first two derivatives, and is initialized to zero; $A = [1 \text{ dt} \text{ } -\text{dt}; 0 \text{ } 1 \text{ } 0; 0 \text{ } 0 \text{ } 1]$. This assumes the filter has a coordinated start with the flight controller when the drone is at rest and assumes a balanced start position. The prediction of the error covariance matrix, P , is initialized as: $P = [1000 \text{ } 0 \text{ } 0; 0 \text{ } 1000 \text{ } 0; 0 \text{ } 0 \text{ } 1000]$. The KF recursively updates the values of P to within reasonable limits from the initially high values. The scaling matrix, B , of the control noise is assumed to be zero for the purposes of simulation. P is then updated with consideration of values of A , according to the following Matlab script: $P = A * P * A'$. Filter observations, z , are then equated to the block input signal, u , which is the angle measured by the drone. The matrix, C , that maps the

current drone state to the measurements read by the IMU, which in this case is set to: $C = [1; 0; 0]$, since the filter estimates only the angle and other derivatives are not needed. Calculation of the prior state estimate is then performed using Equation 16, with x initialized to 0; $x = [0; 0; 0]$. The filter gain is implemented in a two-step process, by applying Equation 17 to determine S , the denominator of the gain equation, and then Equation 18 to calculate gain K . R is a matrix that contains the sensitivity values of the sensors used to implement the KF, however for simulation R is initialized to zero since the simulated IMU signal has no sensitivity; $R = [0 \text{ } 0; 0 \text{ } 0]$. Finally, the filter finds the posterior state estimates (x) and updates the error covariance matrix (P) using Equations 19 and 20 respectively. The output of this block is the estimated angle by the KF based on the inputted measured angle from the flight controller.

$$y = z - C x x$$

Equation 16

$$S = C x P x C' + R$$

Equation 17

$$K = P x C' / S$$

Equation 18

$$x = x + (K x y)$$

Equation 19

$$P = (I - (K x C)) x P$$

Equation 20

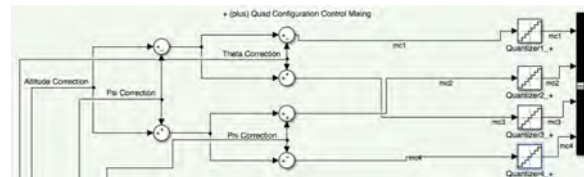


Figure 8. Simulink model of the control mixing signal processing for Quadcopter Plus Configuration.

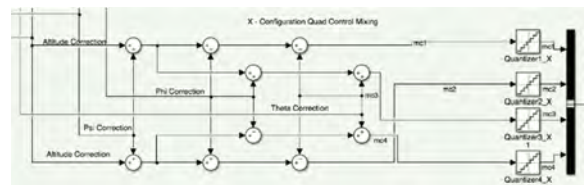


Figure 9. Simulink model of the control mixing signal processing in Ex Configuration.

A control signals mixer has been implemented to provide functions that scale and map the input signal to one that is a meaningful command to the actuator, such as correction signals are translated into throttle commands per motor to drive the quadcopter to the required position. To provide the mixing equations for each of the two configurations



Plus and Ex: changes in speed and required rotational direction of the four propellers map to changes in elevation, pitch roll and yaw. This control mixer varies from existing solutions due to the ability to utilize both Plus and Ex configurations. Further, this model enables dynamic in-flight configuration toggling which widens design capabilities for drone path planning systems and provides a greater range of possible maneuvers, particularly suitable for indoor environments. Motor control signal mixers use the angles correction signals, which have been calculated by the tracking controller as shown in Figures 8 and 9. Quantizer blocks are used to quantize the mixer control signals since real-time processing is a discrete process.

Drift Correction: Lateral drifts have adverse effects on quadcopter indoor SLAM, path planning and control techniques due to drift calculation complexity. Drifts experienced indoors are primarily due to reflected air turbulences from obstructions surrounding the drone such as furniture. EKF algorithms provide estimation of the state of non-linear systems suffering from highly unpredictable noises [16]. In this system, EKF is applied for the drift correction algorithm, illustrated in Figure 10, to estimate the actual pitch and roll of the drone as abstract from the control process. This is also used as feedback signal for the flight controller angle estimator. These are subtracted from the input command signals and their difference estimates the error representing quadcopter drift, which is corrected automatically by the flight controller. The gyroscope and accelerometer have been simulated for calculation of drone pitch and roll angle without interference of noises from the control processes.

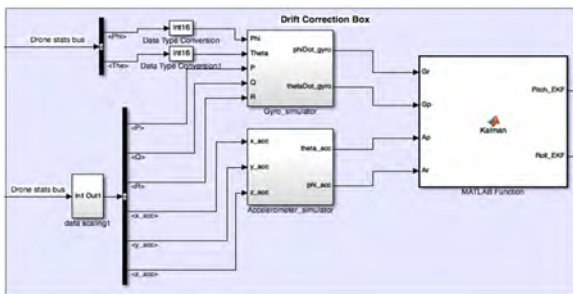


Figure 10. Drift Correction implementation on Simulink: pitch and roll measurement readings of the drone are input to the EKF filter which then estimates the actual angles. The estimated angles are fed back to the flight controller where they are subtracted from the desired angle to give a new angle correction signal that compensates for the non-linear drifts.

Fuzzy Logic Controller: The flight controller has a configuration and several modes that provide improved stability performance over existing systems [2], [14], based on user preferences, yet as flight conditions change and command inputs vary, controller stability performance can degrade. To compensate for these stability variances experienced during changes in flight environment and issued commands, a FLC has been designed and implemented to provide overall improved controller stability performance. During high disturbance greater demands on drift correction are required and the flight controller achieves better flight stability using the Extra Stable mode. When requiring fast and sharp movements the Aerobatic mode outperforms its counterpart. Due to such demands, commanded pitch and roll angles provide function in determining the flight tracking controller mode selected. Further, Ex configuration is able to sustain greater forces due to disturbance than the Plus configuration and as such the former configuration is used to attain greater flight stability in high disturbance conditions. A Mamdani model for the FLC is implemented in Simulink (Figure 11), with commanded pitch and roll angles as input to determine the flight tracking controller mode. In addition, the disturbance is input in the form of values within the covariance matrices of the drift correction EKF to determine the required quadcopter configuration. The four membership functions of pitch and roll inputs are the same; this function is shown graphically (Figure 12 and Figure 13). Membership is assigned based on the current pitch or roll angle that varies from -90 to 90 degrees; if the angle is between -90 and -45 or 45 and 90 it is characterized as high, otherwise it is awarded low. There are two membership functions for disturbance, as shown graphically (Figure 14); if the disturbance is less than 20% of the maximum EKF noise covariance values it is labelled low, while if the disturbance is greater than this value (20%) it is categorized as high.



Figure 11. Overall design of the FLC of the flight controller.



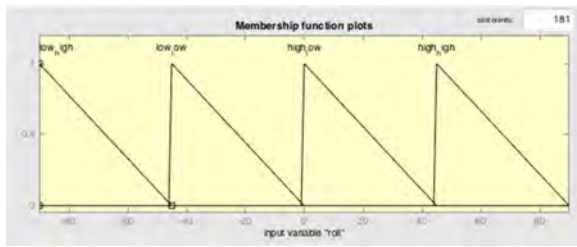


Figure 12. Membership Functions of the Roll input.

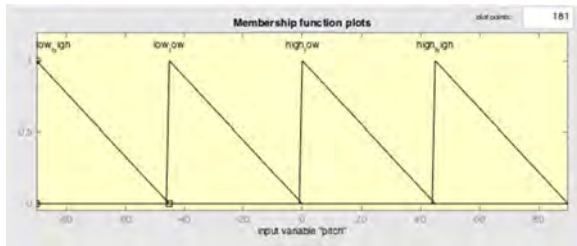


Figure 13. Membership Functions of Pitch input.

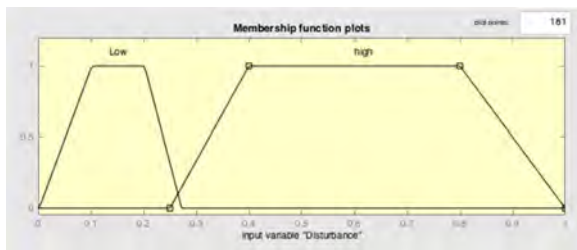


Figure 14. Membership Functions of the Disturbance input.

3. Results

System design and implementation for quadcopter control has been implemented and tested in Simulink and Matlab. System position control is achieved using the Tracking Controller and Mixing Controller. Noises attributed to measurement and control processes are minimized by the KF and Complementary Filter. Flight drifts are calculated by the EKF which are then compensated by the Tracking Controller. The FLC establishes the correct flight mode and drone configuration for optimal stability and dexterity performance amid varying environmental conditions (disturbances, input commands). Results were collected and analyzed from the Position Controller, Tracking Controller with KF, Complementary Filter of the IMU Output, Drift Correction using EKF and FLC Controllers. Performance is evaluated in terms of accuracy, flight stability and error minimization.

Position and Tracking Controller: Controllers were subject to testing during drone flight path change to determine the behavior of the first part of the position controller, in which the desired position is mapped into drone's body frame velocity to calculate the error in the body frame. Figure 15 shows the commanded x-coordinate (in blue) and current x-coordinate measurement (in red), in addition to the output of the position controller (in yellow) for velocity of the body-frame along the x-axis. Testing carried out on the second stage of the controller where error in the body-frame velocities is translated into a desired angle. Figure 16 shows the commanded Theta angle represented (in red) and measured Theta angle (in

blue) captured from the test. Test results for the tracking controller in Aerobatic Mode are shown in Figure 17 with the commanded Theta (pitch) angle (in blue), the correction signal (in yellow), and the measured pitch (in red). Figure 18 shows test results for the tracking controller in Extra Stable Mode with the commanded Theta angle (in blue), measured Theta angle (in red) and correction signal (in yellow).

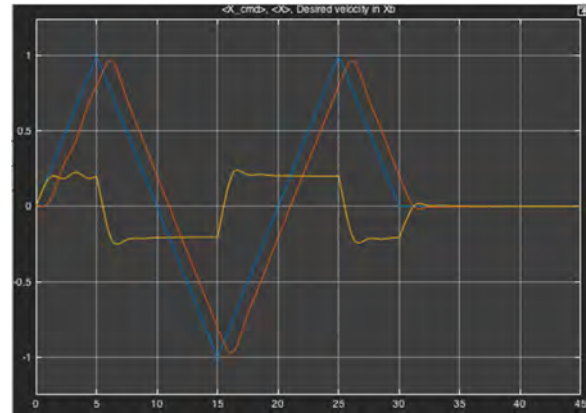


Figure 15. Commanded signal (blue), current measurement (red) and output of the Position Controller for body-frame velocity (yellow); Time versus Position (Radians).

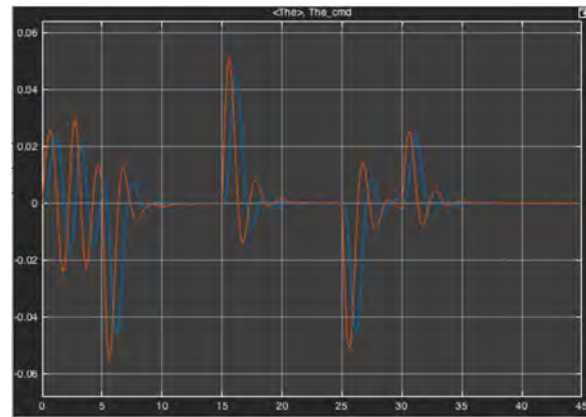


Figure 16. Output of the second stage of the position controller: commanded (red) and measured (blue) signal; Time versus Position (Radians).

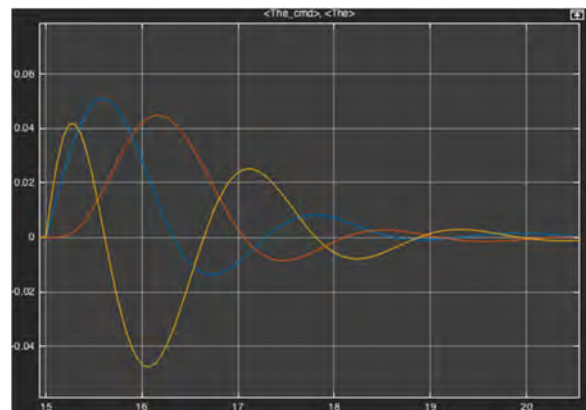


Figure 17. The Theta correction signal of the Aerobatic mode (yellow), the commanded Theta signal (blue) and measured Theta (red); Time versus Position (Radians).



Results captured in Figures 15, 16, 17 and 18 show stable position and tracking controller performance. Drone angles followed the commanded values driven by the correction signals. These were successfully mapped into the proper form of input at the different levels of the controller. Results in Figure 17 reveal that the Aerobatic mode of the tracking controller has a stable bounded output and the drone is following the fast changing theta commands, however, it has a small time lag that could be reduced with further tuning of the PID parameters. Results in Figure 18 related to the Extra Stable mode, show that the controller successfully rejects the control noises present in the time range [2; 4.5]. It is possible to observe that in the same time range the Theta angle is not affected by noise. The noise rejection is a result of using the KF inside this mode. Although the controller caused the drone to exceed the command values when the time is equal to 0.8, it manages to correct this minimizing the difference with the commanded signal. This issue is a result of the KF error covariance matrix which provides less accurate state estimates at the first iterations of the KF. Analysis of the controllers reveals stable and accurate performance, meeting design objectives.

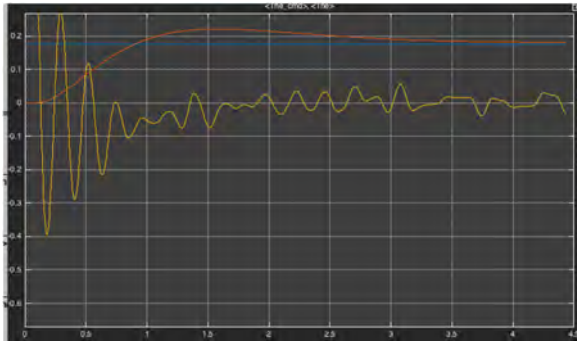


Figure 18. Output showing behavior of the Extra Stable mode of the tracking controller; Time versus Position (Radians); Time versus Position (Radians).

Drift Correction: Sample test results of the EKF filter for drift correction are shown in Figure 19. The graph is a result of simulating two consecutive harsh maneuvers on the Pitch rotational axes, with results of measured pitch angle, estimated pitch angle and their difference. This test is intended to validate EKF design and for the completion of the drift correction process by feeding back the calculated drift to the following stage of position controller. Figure 20 shows the simulation results for drone trajectory as a function of the entire flight controller: in the presence, and absence, of the drift correction mechanism. Figure 20a (left-most graph) represents the actual drone path (blue circles) versus the commanded path (red line) with drift correction active, while Figure 20b reveals drone path and commanded path for drift correction set to inactive.

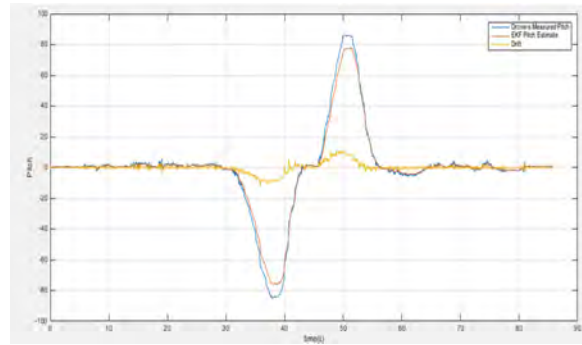


Figure 19. The behavior of the EKF for drift correction: these results show estimated pitch (in blue), measured pitch angle (in red) and the difference between these (in yellow).

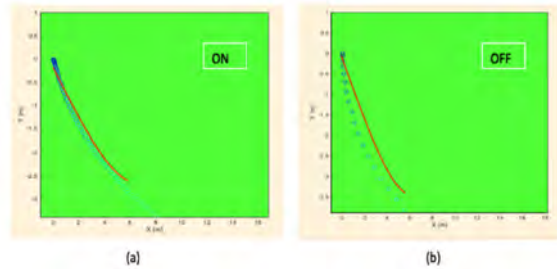


Figure 20. Simulation results of the output of the flight controller trajectory (planned path in red, actual path in blue) in the x-y plane with the presence of simulated wind disturbance, with (a) and without (b) drift correction.

FLC and System Results: Results of the FLC are generated in a simulation-based environment, utilizing functions within the Fuzzy Logic Toolbox embedded in MATLAB. Sample results representing the output model for variation in roll and pitch angle are shown in Figure 21. Sample results representing output configuration for variation in roll and pitch angle are visualized in Figure 22, and with variation in percentage of disturbance in Figure 23.

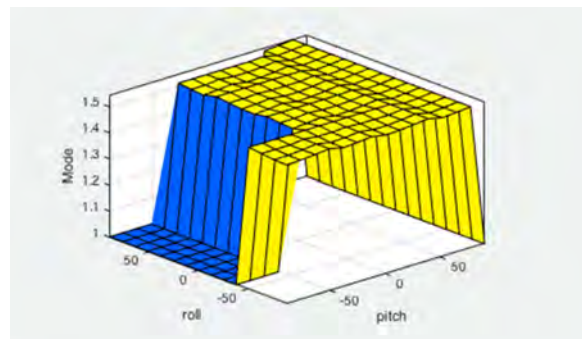


Figure 21. Results showing output "mode" as a function of variations in roll and pitch angles.



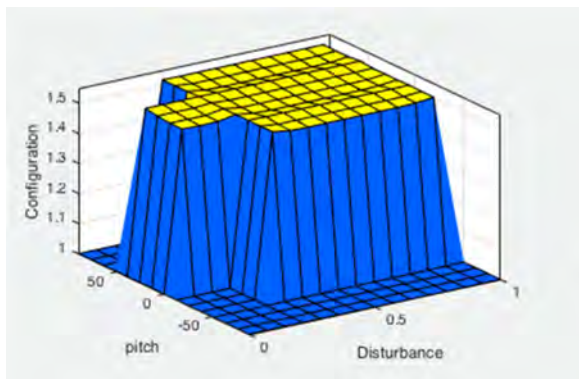


Figure 22. Results of output "configuration" as a function of variations in roll and pitch angles.

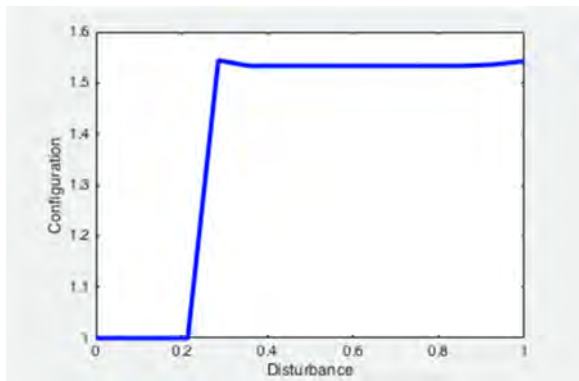


Figure 23. Results of output configuration based on the percentage of the disturbance.

4. Discussion

Position and Tracking Controller: The function of the Position Controller and Tracking Controller is to calculate and generate the correction control signals for the UAV. The Tracking Controller is designed to operate with two modes, wherein each mode should produce a correction signal in terms of the rotational angles, pitch, roll and yaw, as well as the altitude of the drone. As is evident in the results presented, this functionality is achieved. Both controllers exhibit fast, accurate and stable output for drone flight control correction, with the second mode exhibiting greater stability due to incorporation of a KF that increases its reliability in the presence of noise; appropriately minimizing the impact of this noise on flight control.

It is noted that Position Controller output is input to the tracking controller. Both controllers utilize the PID control technique and its customization. The controllers prove effective in determining the desired rotational rates when the input data are x, y, and z coordinate points. The output of the first stage is the desired correction in velocity of the body frame. The correction signal is adequately applied to the entire flight controller and enables successful control in the x, y, and z state coordinates of the drone. In the second stage of the controller, desired rotational rates are effectively input, along with the correction values of the first stage, in addition to measured angles of the drone. This resulted in appropriate, dynamic changing of the angle of the drone in accordance with the desired rates of

change. Both stages of the Position Controller, in addition to the Tracking Controller, exhibited correct, responsive behavior for UAV flight control correction. The implementation of multiple flying modes of a robust controller widens the capabilities of a UAV by enabling it to fly and maneuver in challenging environments, as opposed to mono-mode flight controllers which require manual tuning for significant changes in flight capability and control technique.

Drift Correction: Drifts, caused by disturbances due to obstructions, are one of the major challenges faced by autonomous quadcopters during indoor flight. Non-linear lateral drifts are approximated in this work using a prediction model; EKF is applied in this capacity to provide an optimal solution for state estimation of this non-linear system of drifts. EKF is implemented for this stage of the flight controller for drift estimation and in response, to correct the UAV behavior after encountering such drifts. In this application, the EKF proved effective in estimating the pitch and roll angles of the UAV as isolated from the interference caused by the other controller in order to avoid the effect of additive noise. The state estimations of the EKF differed from the filtered measured angles and the difference between the two values was considered to be the estimated drift. As evident from the results, the implementation of the EKF provides a close estimate for the pitch and roll angles to those of the measured. This scenario was conducted in a simulated environment that did not introduce extreme surrounding environmental noise and as such did not cause significant drifts.

The expected behavior of the EKF is validated with the results obtained, as shown in Figures 19 and 20. Measured angles by the drone still contain process and control noise and as such have a larger magnitude than the values estimated by the EKF. However, this difference is sufficiently small. Overall, the drift correction mechanism reveals its capability of generating effective compensation for the lateral drifts: with an actual drone path suitably following the commanded (planned) trajectory (Figure 19b).

FLC: The flight controller has two flying modes and two kinematic configurations. The performance of each option of selected mode and configuration varies based on the input data and the flight environment. Due to this variability there is a need for another layer of control with the function to determine the flight mode and configuration of the UAV. The FLC was developed to provide this function. The FLC has three input signals of pitch, roll and disturbance, and two output values: mode and configuration. The FLC utilizes a set of 23 rule to determine the outputs based on the given inputs. Results show that by introducing the FLC greater flight stability is achieved by enabling selection of the most appropriate mode and configuration. The addition of the extra layer of control provides further self-governing capabilities, in which the FLC decides the suitable flight mode based on drone sensor readings and measured noise signals from the other controllers. The FLC controller successfully integrates the efforts of the other developed controllers, over existing methods [2], [14], [16].



5. Conclusions and Recommendations

A hybrid flight controller has been designed and tested in a simulation environment that achieves a high degree of stability and dexterity in maneuverability within an indoor location that is subject to non-linear drifts. For the Position Controller, a KF is applied within the mode that offers high in-flight stability, effectively rejecting induced noise and subsequently reducing the errors associated with Controller processing. Flight Control modes implemented achieve sound results in terms of stability within extreme disturbance conditions, in addition to sharp, fast, and accurate maneuvers under normal conditions. The Motor Control Mixer effectively maps the angle correction signals into a required throttle for each of the four motors, enabling flight in two separate configurations, dependent on desired throttle output. Non-linear drift estimation is effectively implemented using the EKF, in which compensation commands by the Position Controller are executed based on estimated drift calculation for pitch and roll angles. In addition to the PID and associated derivatives for motor control, the FLC proves useful in offering a further layer of flight control stability.

The proposed system can be further improved: by adding further Flight Controller modes for selection to enable greater specificity in flight behavior for different flight conditions. An automatic PID and PIDD² fine tuning mechanism may help to achieve even greater results in terms of system stability. Sensor fusion techniques, Bayesian Network and Central Limit Theorem, may be utilized in the IMU to achieve higher accuracy in angle measurements, which may lead to better state estimates of rotational angles and rates. KF and EKF design may be extended to include yaw and altitude UAV states, requiring addition of magnetometer and barometer sensors to attain these measurements. The FLC may be extended to process greater numbers of input, output and rule sets to provide more heightened control over the Flight Controller, particularly if the addition of Flight Controller modes is realized.

6. References

- [1] M. Leichtfried, C. Kaltenriner, A. Mossel, H. Kaufmann. Autonomous Flight using a Smartphone as On-Board Processing Unit in GPS-Denied Environments, *Proceedings of International Conference on Advances in Mobile Computing and Multimedia*, pages 341-350, 2013.
- [2] A. Sharma, A. Barve. Controlling of Quad-Rotor UAV using PID Controller and Fuzzy Logic Controller, *International Journal of Electrical, Electronics and Computer Engineering*, vol. 1, issue 2, pages 38-41.
- [3] P.E. Pounds, D.R. Bersak, A.M. Dollar, Stability of Small-Scale UAV Helicopters and Quadrotors with Added Payload Mass under PID Control, *Autonomous Robots*, vol. 33, issue 1-2, pages 129-142, 2012.
- [4] P. Salaskar, S. Paranjape, J. Reddy, A. Shah, Quadcopter-Obstacle Detection and Collision Avoidance, *International Journal of Engineering Trends and Technology*, vol. 17, issue 2, pages 84-87, 2014.
- [5] S. Bouabdallah, Siegwart R, Full Control of a Quadrotor, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 153-158, 2007.
- [6] A.L. Salih, M. Moghavvemi, H.A. Mohamed, K.S. Gaeid, Flight PID Controller Design for a UAV Quadrotor, *Scientific Research and Essays*, vol. 5, issue 23, pages 3660-3667, 2012.
- [7] L. Heng, L. Meier, P. Tanskanen, F. Fraundorfer, M. Pollefeys, Autonomous Obstacle Avoidance and Maneuvering on a Vision-Guided MAV using On-Board Processing, *Robotics and Automation (ICRA)*, 2011 IEEE international conference on, pages 2472-2477, 2011.
- [8] J.C.V. Junior, J.C. De Paula, G.V. Leandro, M.C. Bonfim, Stability Control of a Quad-Rotor using a PID Controller, *Brazilian Journal of Instrumentation and Control*, vol. 1, issue 1, pages 15-20, 2013.
- [9] B. Kada, Y. Ghazzawi, Robust PID Controller Design for an UAV Flight Control System, *Proceedings of the World Congress on Engineering and Computer Science*, vol. 2, issue 1-6, 2011.
- [10] L.C. Chien, Fuzzy Logic in Control Systems: Fuzzy Logic Controller- Part I, *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, issue 2, pages 404-418, 2011.
- [11] I. Sa, P. Corke, Estimation and Control for an Open-Source Quadcopter, *Proceedings of the Australasian Conference on Robotics and Automation*, pages 7-9, 2011.
- [12] M. Achtelik, T. Zhang, K. Kühnlenz, M. Buss, Visual Tracking and Control of a Quadcopter using a Stereo Camera System and Inertial Sensors, *Mechatronics and Automation, IEEE, ICMA International Conference*, pages 2863-2869, 2009.
- [13] M. Hehn, R. D'Andrea, Real-Time Trajectory Generation for Quadcopters, *Robotics, IEEE Transactions on*, vol. 31, issue 4, pages 877-892, 2011.
- [14] A.K. Gaurav, A. Kaur, Comparison Between Conventional PID and Fuzzy Logic Controller for Liquid Flow Control: Performance Evaluation of Fuzzy Logic and PID Controller by using MATLAB/Simulink, *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 1, issue 1, pages 84-88, 2012.
- [15] Y. Al-Younes, M.A. Jarrah, Attitude Stabilization of Quadrotor UAV using Backstepping Fuzzy Logic and Backstepping Least-Mean-Square Controllers, *Mechatronics and Its Applications, ISMA 5th International Symposium*, pages 1-11, 2008.
- [16] C.N. Hamdani, E.A.K. Rusdhianto, D.E. Iskandar. Perancangan Autonomous Landing Pada Quadcopter Menggunakan Behavior-Based Intelligent Fuzzy Control, *Journal Teknik ITS*, vol. 2, issue 2, pages E63-E68, 2013.
- [17] L. Meier, P. Tanskanen, F. Fraundorfer, M. Pollefeys, Pixhawk: a System for Autonomous Flight using Onboard Computer Vision, *Robotics and automation (ICRA)*, 2011 IEEE international conference on, pages 2992-2997, 2011.



Biographies



Dr. Catherine Todd received her B.Eng. (Electrical) and PhD. (Electrical) from the University of Wollongong, Australia. Currently she is working as an Assistant Professor of Engineering at Hawaii Pacific University. Her research interests include

medical image processing, control systems and real-time simulation.



Dr. Stefano Fasciani is currently an Assistant Professor of Engineering at University of Wollongong in Dubai. He is doing research work in sound engineering. His main research interests are in voice processing,

noise cancellation, HCI, acoustic and electronic engineering.

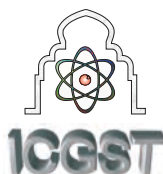


Mr. Hussien Koujan is working in Industry as an Electrical Engineer at Altorath International Engineering Consultants, having graduated from the University of Wollongong in Dubai with a Bachelor of Engineering

(Electrical).







Applying Hyper-Fuzzy Extended Kalman Filter to Indoor Security Monitoring

Ali Alqudaihi, Mohamed Zohdy Ph.D. Hoda Abdel-Aty-Zohdy Ph.D.
Department of Electrical and Computer Engineering, Oakland University, Rochester, MI, USA
 aaalquda@oakland.edu, zohdyma@oakland.edu, zohdyhsa@oakland.edu

Abstract

This paper presents a new monitoring security system, based on data fusion (DF) from three heterogeneous sensors which are pressure (PR), active sonar (AS) and infrared (IR). Sensors similarity and complementarity concepts are applied after proper data in real-time. Data is then fed to a Fuzzified Extended Kalman Filter (FEKF) which perform the data fusion. Based on the risk/suspiciousness degree of the monitoring system, moving agents are assessed. A Hyper-Fuzzy logic is utilized in the system as a decision making sub-system that allows the monitored area to be carefully and completely checked for any abnormal activities. The system is then animated to show how the fusion and visualize monitoring.

Keywords: *Multi-Sensor Data Fusion, Feature Extraction, Fuzzified Extended Kalman Filter, Hyper-Fuzzy, Similarity and Complementarity, Measurements Fusion, State Vector Fusion.*

Nomenclature

DF	Data Fusion
PR	Pressure Sensor
AS	Active Sonar
IR	Infrared Sensor
KF	Kalman Filter
EKF	Extended Kalman Filter
FEKF	Fuzzified EKF
AD	Arrival Direction
AT	Arrival Time
MF	Measurement Fusion
SVF	State Vector Fusion
FL I	Fuzzy Logic I
FL II	Fuzzy Logic II
HF	Hyper-Fuzzy
MaN	Number of Moving Agents
AsV	Value of the Assets
AdS	Distance from Assets
SD	Suspiciousness Degree

1. Introduction

In the past years, the increase of buildings and homes breaches grabbed the attention of governments and researchers to develop more reliable security monitoring systems that can detect the behaviors of intruders and take

action. Multi-Sensor fusion plays an important role in modern security systems which can provide more robust and capable data processing that wouldn't be possible with the use of systems that are built based on a single sensor data. [1]

The principle of multisensor measurements fusion has also been used in military applications for a long time to achieve precise target tracking and recognition. Similar techniques and models are used in medical applications and robotics to enhance the quality of their monitoring and guidance. Multiple types of sensors are usually combined to improve estimations based on appropriate algorithms that can maximize their accuracy. [6,7]

Designing such systems requires proper selection of homogeneous sensors types and processing techniques since the fused provided data can lead to unreliable estimations and consequently poorer decisions by the system which is not acceptable in most cases. Another advantage of fusing different sources of data is the capability of blending accurate and inaccurate measurements that are delivered from sensors with unequal variances and have unknown biases [6].

Similarity and complementarity principles are applied in this paper to pre-process the raw sensors information before they are fused to prevent the system from making imprecise conclusions. The two processes will help to pass the only consistent data and reject the untrustworthy sensors measurements that contain undesirable noisy data which will help the monitoring system to make reasonable judgements of the environment under surveillance [3].

The extracted features from sonar and infrared are combined to serve as one virtual data that represents the position and velocity of any moving object. The pressure sensors floor array will assist the monitoring system to locate the moving agents providing their weights and location. The weights are important to recognize each person with his/her distinctive weight where it's assumed that no person share the same body weight with others. Raw data from sensors are not complete and may not reflect the real environment being observed since each sensor shows part of the truth. Fusing all these sensors data together will result in more reliable observation system that shows the bigger picture with proper details, which can then be utilized to understand the different behaviors of the agents under supervision.



In section 2, a survey of related work is given. Section 3 provides the system's sensors functionality and their roles in monitoring the area under observation. In sections 4-7, methods of sensors selection, preprocessing, and fusing are provided. Section 8 discusses the EKF and how it is Fuzzified applying Fuzzy Logic. Section 9 shows the system dynamic model; while, section 10 examine the Hyper-Fuzzy decision method. In section 11, the system animation is explained. And finally, section 12 an investigation and discussion of the monitoring system results are presented.

2. Related Work

Many papers and researches have been accomplished on multisensor data fusion applying several methods and approaches. This paper is not the first of its kind; however, the type of sensors, sensors selection, filtering, and decision making present an altered combination methodology as a new way of tackling the same goals in addition to the classification of the target being tracked and monitored. For example, in [3] the laser and radio frequency were the major sensors fused to achieve a similar goal as this paper intend to do. In addition, the filtering process was based on EKF alone without the aid of Fuzzy Logic as a technique of improving the estimation. Another work was done employing image and acoustic radar sensors [6] as the two sensorial modules that were fused to detect the intruders in the covered zone. Neural network algorithm was applied to detect the moving agents and fire an alarm in case the active breaching.

The most recent challenges and development of data sensor fusion are addressed in [18] where they discussed the limitations of the multi-sensor fusion applications and their advantages. Some advances were in algorithms classification and fusion methods such as wavelet-based and neural network fusion. In remote sensory data fusion, their work explains the identification, classification, and detection methods of objects applied to fusion of images. The suggested methods in [18] may improve the capability of image sensor fusion by enhancing the fusion algorithms and refining the quality of assessment performance.

Sensor Fusion as an application has found its way in navigation system utilizing GPS, inertial sensors, and vision sensors that are currently hot topics in automation industry. The orientation of the vehicles and robots as well as the position estimation in real-time has enhanced with more heterogenous sensors data that being fused to achieve those goals as was investigated in [19,20].

3. The Sensors

This anticipated system model harness the advantages of three types of sensor arrays to realize the environment under surveillance by tracking and detecting the quantity of agents and their locations with respect to the assets positions. First, the pressure sensor array will provide the number of moving and stationary objects associated with their different weights and pressure on the monitored floor. Then, a group of AS and IR sensors will start tracking each agent position and velocity.

A. Sonar Sensor

Sonar acronym is derived from SOUNd Navigation And Ranging by Frederick Vinton Hunt in 1942 which refers to both passive and active acoustic range finding devices. [12] Passive sonars measures direction and distance of the noise that is made by the tracked creature or machine without transmitting any signal. Alternatively, in active sonar (AS) a sound pulse is emitted by the transmitter and then received after a period of time "delay" by the receiver. The receiver in AS obtain the arrival time (AT) and the arrival direction (AD) for every pulse transmitted by measuring the time difference between both transmitted and received pulse [11,13]. Figure 1 explains the AS components and principle.

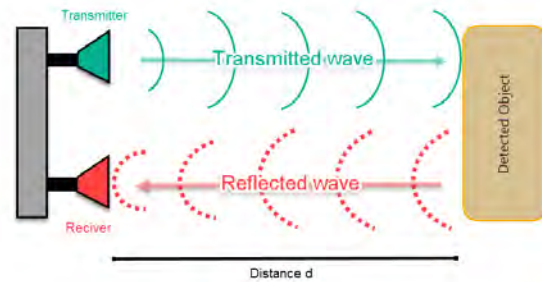


Figure 1: Active Sonar Components and Principle

Equation (1) shows the how the distance of a tracked object is calculated by multiplying the time difference between the outgoing and incoming echo and the speed of light divided by 2 [5,11,13]:

$$d_k = \frac{c \cdot \Delta t_k}{2} \quad (1)$$

AS distance measurement is affected by the temperature and pressure of air, and the chemical structure of air components; however, AS are not tolerated by the other distractions like smoke, light brightness, and electromagnetic fields interventions. [5,12]

B. Infrared Sensor (IR)

Infrared refers to the ability to emit the electromagnetic radiation toward a target to detect the temperature and the motion. The emitted radiation from IR sensors is invisible to the human's eye but it can be received by a photodiode that is sensitive to the infrared light and have the same wavelength of the emitter diode. IR is famous for real time tracking applications like the proposed security system.

Although IR sensors are reliable for close range distance measurements, they also provide a good approximation of the long range object detection. [5]

IR sensor fit the purpose of this proposed security monitoring system since it is capable of detecting any living creature and any lifeless object. IR photodiode senses the emitted infrared light from far distances in wide areas. There are many types of IRs based on their output such as voltage or digital output. Infrared sensor detection accuracy can be distracted by the temperature of the room and the temperature of the tracked objects/agents. [6,14]



C. Pressure Sensors Floor Array

One of the highly effective sensors that is barely used in security systems is the pressure sensor. Floor integrated pressure sensitive system is utilized in this proposed security monitoring model to obtain the weight and location of any moving or stationary objects/people. The monitored area is completely covered with pressure tiles to fulfil this purpose. Every stationary object is marked in the system with its respective weight and location; so that the monitoring system is able to recognize whether the valuable assets has been moved or stolen and to distinguish between what or who needs to be monitored. Once an agent or a guard step on any part of the floor, the system will start measuring and tracking his/her steps pressure and altitude on the monitored surface [7]. The monitored area is assumed to be all covered with pressure sensors tiles as the dotted lines in Figure 2.

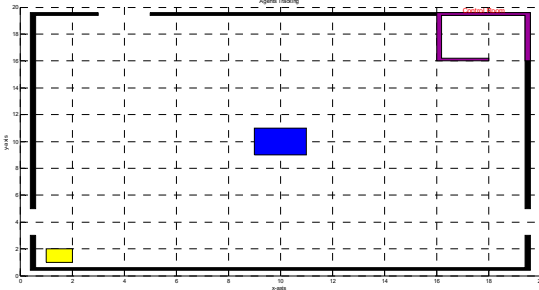


Figure 2: Pressure floor array as animated in the proposed system “Matlab”

4. Preprocessing and Sensors Selection

Since there are three types of sensors (Sonar, IR, and Pressure) in the monitoring model; three Sonars and three IRs are assigned for each agent in the monitored area. The raw data collected from each sensor within the same group need to be fused before the fusion of the three categories to obtain one measurements stream from each type. This preprocessing will help the system to blend more reliable data. Measurements Fusion (MF) is applied to blend all the sonars data together and the IRs together to end up with only one data stream for each category. For the pressure sensors, it is consisted of floor array and it's impossible to stack more than one tile on each other to have multiple tiles measurements.

5. Sensors Complementary

In this multi-data fusion system, sensors need to communicate and collaborate dynamically to deliver a clear full picture of the environment under surveillance. All individual uncertain geometric sensory data have to alternate their vision of the agents with each other in a way that it tells the system the current situation, where sometimes one sensor has a missing or misleading measurements that lead to unreliable decision making. [3,10]

6. Similar Sensors Fusion “Sensors Similarity”

The replicated Sonars and IR's sensors data are combined into one sensory data; where each sensor has a different

view of the environment under surveillance and its components. While other sensor similarity techniques sacrifice some sensory data in order to select the similar/close range data as in [3, 5]; fusing the sensors by applying Measurement Fusion method can harvest the advantages of all sensors in the monitored environment.

Measurements Fusion (MF) is the first prominent method of blending sensors that have similar properties and functionality. Unlike State Vector Fusion (SVF), MF can blend the data prior to the estimation process. [9,15]. Measurements vectors Z_k^1 , Z_k^2 and Z_k^3 from the three Sonars and their associated measurements' noises “ R_s^1 , R_s^2 and R_s^3 ” are combined into a single new vector Z_s^4 recursively.

For two sensors measurements (Z_k^1 and Z_k^2) and their errors (R_s^1 and R_s^2) we may find the resulting combined measurement vector applying MF as following:

$$Z_k^{1,2} = Z_k^1 - (R_s^1 * (R_s^1 + R_s^2)^{-1} * (Z_k^2 - Z_k^1)) \quad (2)$$

$$R_k^{1,2} = ((R_k^1)^{-1} + (R_k^2)^{-1})^{-1} \quad (3)$$

However, for three sensors measurements $Z_k^{1,2,3}$ is formulated by replacing Z_k^1 with Z_k^3 and substituting $Z_k^{1,2}$ into Z_k^2 . Then the three sensors measurements Z_k^1 , Z_k^2 and Z_k^3 , as in this proposed monitoring system, the MF equations are computed as:

$$Z_k^{1,2,3} = [(R_s^1 * R_s^2 * Z_k^3 + R_s^1 * R_s^3 * Z_k^2 + R_s^2 * R_s^3 * Z_k^1) * [R_s^1 * R_s^2 + R_s^1 * R_s^3 + R_s^2 * R_s^3]^{-1}] \quad (4)$$

And for the measurements errors covariances become:

$$R_k^{1,2,3} = ((R_k^1)^{-1} + (R_k^2)^{-1} + (R_k^3)^{-1})^{-1} \quad (5)$$

After applying MF to three homogenous sonar sensors we get the fused MF as in Figure 3.

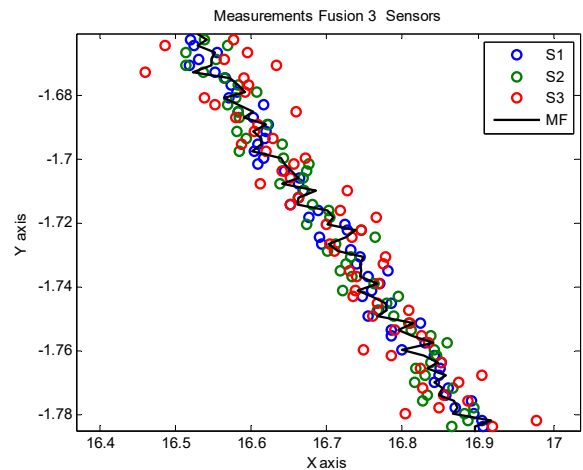


Figure 3: MF of three sonar sensors

The resultant fused measurements and their noise covariances are then filtered through separate Kalman Filters where the State Vector Fusion (SVF) will be applied later.



7. Sensor Fusion Applying State Vector Fusion Method

Unifying the sensory data of the same type of sensors in the preprocessing phase applying MF is the first fusing stage; however, the resulting combined data from the three types of sensors are then fused altogether utilizing the State Vector Fusion (SVF) technique. For two sensors to be fused by SVF, the MF fused measurements for the same type of sensors is fused with the second type of sensor as following [5,15]. Pressure sensor state vector ($\hat{x}_{k-1}^{prs}$) is fused with IR sensor state vector ($\hat{x}_{k-1}^{IR}$) first equation (6) and then the AS sensor state vector ($\hat{x}_{k-1}^{AS}$) is fused with the resulting first vector equation (8) where the new estimate is given by the following equation:

$$\hat{x}_{k-1}^1 = \hat{x}_{k-1}^{prs} + P_{k-1}^{prs} [P_{k-1}^{prs} + P_{k-1}^{IR}]^{-1} [\hat{x}_{k-1}^{IR} - \hat{x}_{k-1}^{prs}] \quad (6)$$

Whereas, P_{k-1}^{prs} and P_{k-1}^{IR} are the error covariance matrices of both pressure and IR sensors respectively. Their combined covariances can now be stated as in equation (7):

$$P_{k-1}^1 = P_{k-1}^{prs} + P_{k-1}^{prs} [P_{k-1}^{prs} + P_{k-1}^{IR}]^{-1} P_{k-1}^{prsT} \quad (7)$$

The AS fusion with both previous combined sensors is computed by equations (8,9) as:

$$\hat{x}_{k-1}^2 = \hat{x}_{k-1}^{AS} + P_{k-1}^{AS} [P_{k-1}^{AS} + P_{k-1}^1]^{-1} [\hat{x}_{k-1}^1 - \hat{x}_{k-1}^{AS}] \quad (8)$$

$$P_{k-1}^2 = P_{k-1}^{AS} + P_{k-1}^{AS} [P_{k-1}^{AS} + P_{k-1}^1]^{-1} P_{k-1}^{AST} \quad (9)$$

Where the results of all sensory data SVF are then fed into FEKF algorithm to enhance the estimation and tracking accuracy of the monitoring system which will lead to reliable decision that can reach beyond the results of other proposed fusion methodologies.

8. Fuzzified Extended Kalman Filter

Kalman Filter (KF) is a widely-used estimation algorithm in many modern systems which could be found in many engineering applications. In this paper, the nonlinear extension of KF is applied to estimate the states of any moving object based on the data provided from the pressure, sonar, and infrared sensors that are implemented in the monitoring/tracking model. All sensors' data are fused using EKF which allow to increase the accuracy of the system estimation beyond the abilities of KF and EKF alone. All model imperfect components may add unwanted additions to the measurements which are denoted as the measurement and system errors. These errors are characterized as Gaussian white noise [3,8,16]. Extended Kalman Filter similar the Kalman Filter, consists of two iterative stages which are the prediction/projection and update/estimation for the current time instances with no need of storing or processing all the previous predicted measurements; instead, it only requires the latest measurements to estimate the current estimate [19]. Consider the following stochastic model of a nonlinear discrete difference equations as following:

$$\tilde{x}_k = f(\hat{x}_{k-1}, u_{k-1}) + w_{k-1} \quad (10)$$

$$\tilde{y}_k = h(\tilde{x}_{k-1}) + v_{k-1} \quad (11)$$

In Equation (10), the $x \in \mathcal{R}^n$ is a state vector, f is a nonlinear function that links the prior time step $k-1$ to k which is the current time step, and w_{k-1} is a white Gaussian with zero mean system noise. While in equation (11), $y \in \mathcal{R}^r$ is the measurement equation, h is a nonlinear function that links the measurement vector y to the state vector x , and v_{k-1} is a white Gaussian with zero mean measurement noise [17].

A. Prediction Stage

The filter first forecast the current state based on the a posteriori estimation of $\hat{x}$ as in equation (12).

$$\tilde{x}_k = f(\hat{x}_{k-1}) \quad (12)$$

Once the state is predicted, the error covariance P is calculated as in (13):

$$P_k = F_{k-1} P_{k-1} F_{k-1}^T + Q_{k-1} \quad (13)$$

Where F is the Jacobian of $f(\cdot)$ with respect to x is obtained (equations (14,15)) to determine the error covariance which can be computed by taking the partial derivatives of the nonlinear state vector elements to linearize about the mean and the variance of the existing measurements at each time step.

$$F_k = \nabla f_k | \hat{x}_k \quad (14)$$

$$\nabla f_k = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \quad (15)$$

B. Correction Stage

The predicted/estimated states need to be corrected now by calculating additional set of equations that include the Kalman gain K , updated state vector $\hat{x}_k$ and error covariance P_k as following:

$$K_k = P_k H_k^T (H_k P_k H_k^T + R_k)^{-1} \quad (16)$$

Where H is the Jacobian matrix of $h(\cdot)$ with respect to X as in (17,18):

$$H_k = \nabla h_k | \hat{x}_k \quad (17)$$

$$\nabla h_k = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \dots & \frac{\partial h_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_n}{\partial x_1} & \dots & \frac{\partial h_n}{\partial x_n} \end{bmatrix} \quad (18)$$

The current state estimation is then calculated by adding the multiplication of Kalman gain and the innovation vector to the a priori state values:

$$\hat{x}_k = \hat{x}_{k-1} + K_k (z_k - h(\hat{x}_{k-1})) \quad (19)$$



Finally, the error covariance matrix can be updated and adjusted to reflect the change of Kalman gain on the next prediction cycle:

$$P_k = [I - K_k H_k] P_{k-1} \quad (20)$$

C. EKF Fuzzification

Over the years, fuzzy logic technologies have attracted the attention of various groups of people, including researchers and computer scientists. In particular, it has had a significant role in replacing the commonly used forms of technologies in several engineering and scientific applications, especially signal processing, pattern recognition, not to the mention making of integrated circuits. In addition to this, it is employed in other areas such as approximation reasoning. What is more, fuzzy technology is capable of aiding in making decisions besides creating systems that have a considerable capacity to reason well [3,4]. It should be noted that these particular systems were presented in 1965 by Zadeh. Originally, they were meant to compute and adjust data faced with imprecisions or uncertainties [4]. A salient point to note is that the base of all fuzzy systems is the fuzzy logics and sets, which have been designed to imitate brain functionality for uncertain information.

It has already been established that not all problems involving computing and mathematics are easy to go about. As a matter of fact, many of these problems are complex and may seem impossible to compute. Therefore, mathematical tools such as the fuzzy logic are employed to tackle these types of problems.

The above details showed how the known EKF work with respect to the dynamic system and the sensory data which works well for nonlinear models; however, with the aid of Fuzzy Logic I (FL I), the EKF can perform better and provide a more accurate estimation [16]. The resulting combination of FL I and EKF can be termed as Fuzzified EKF or FEKF.

The measurement noise covariances of the sensors are considered constant in most systems that utilize KFs algorithm for estimation and prediction which may prevent KFs from reaching its optimality [16]. The addition of FL to EKF will reduce the estimation error by making the measurement error covariance R and the system error covariance Q variable instead, which in turn can fine-tune the EKF estimation. FL I has the capability of logical reasoning of inaccurate, indistinct, and insufficient information that lead to more adequate decision making. [4]

FL is a technology that use fuzzy sets, a set with soft boundaries, consisting of universe of discourse (X) which is a class in the set theory. FL needs a membership function $\mu(x)$ that allocate a value between 0 and 1 to every single element in the discourse. $\mu(x)$ will ease the reasoning process of FL [3,4,16]. In general, a fuzzy set can be defined by the following relationship:

$$A = \{x, \mu_A(x) | x \in X\} \quad (21)$$

Where A is the fuzzy set of ordered pairs and $\mu_A(x)$ tells to what degree of fuzzy set x belong. For example, if the value of the membership function gets closer to 0, then it is not likely x is a member of the set A ; on the other hand, if the value of $\mu_A(x)$ gets closer to 1, the likeliness of x to be a member of A is very likely [4].

In KF, if the process error covariance Q increase, the gain K will increase trust weight of the innovation vector; however, if the measurement error covariance R increase, the gain K will decrease the trust weight of the innovation vector. Both relationships play a huge role in tuning the EKF. To fuzzify EKF, a new variable need to be introduced (α) which is the weight scalar that will increase or decrease the error covariances of both the process and the measurements [16] where these relationships can be expressed as:

$$Q_k = Q \alpha^2 \quad (22)$$

$$R_k = R \alpha^{-2} \quad (23)$$

The weight scalar α is determined by setting a FL I model that relates the polar coordinates Δr and $\Delta \theta$ discrepancies, and the innovation vector $In_k = (z_k - h(\hat{x}_{k-1}))$ to α .

The FL I rules table is shown below (Table 1) which defines the relationship between the change in angles $\Delta \theta$ and the innovation vector In_k by a transitional variable $C_{In/\theta}$. [16]. Figure 4 and Figure 5 explain the FIS system and fuzzy surface of table 1 relationships and how they are selected subsequently.

$C_{In/\theta}$		$\Delta \theta$		
		S	M	L
In_k	S	S	S	M
	M	S	M	L
	L	M	L	L

Table 1: FL I rules associated with transitional variable

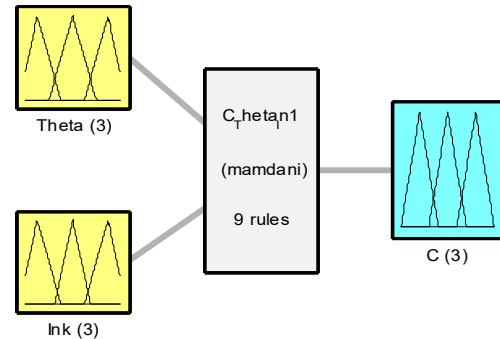


Figure 4: FIS system for transitional variable built by Matlab

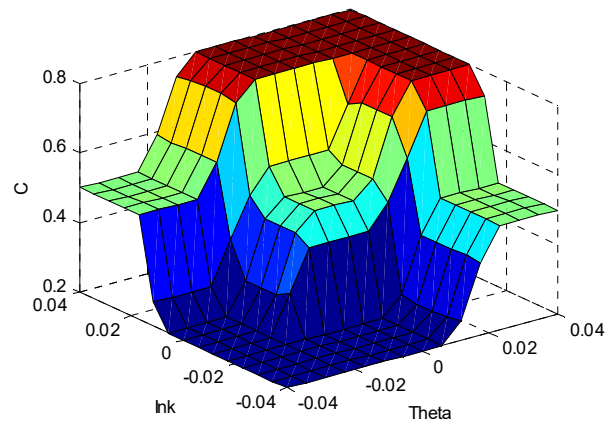


Figure 5: FIS Surface of transitional variable with inputs $\Delta \theta$ and In_k , output $C_{In/\theta}$



This association will then be considered in regard with the distance differences Δr and the weight scalar as shown in Table 2:

α		Δr		
		S	M	L
$C_{In/\theta}$	S	S	M	L
	M	S	S	S
	L	VS	VS	VS

Table 2: FL I rules associated with weight scalar

In the above tables, S stands for small, M stands for medium, L stand for large, and VS stands for very-small value that can be close to zero. Figure 6 and Figure 7 explain the FIS system and fuzzy surface of table 2 relationships and how they are selected subsequently.

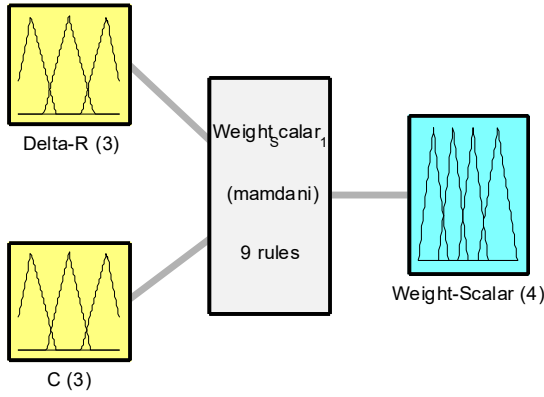


Figure 6: FIS system for transitional variable built by Matlab

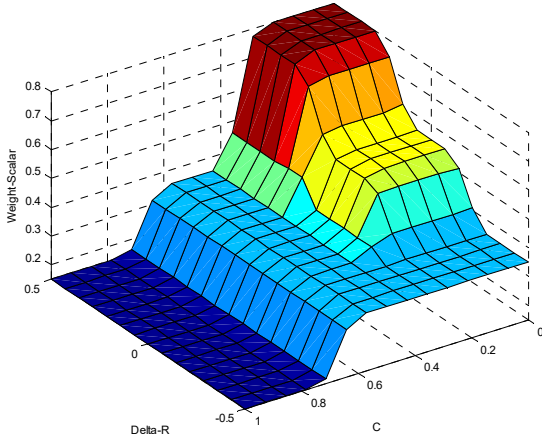


Figure 7: FIS Surface of transitional variable with inputs $\Delta\theta$ and In_k , output $C_{In/\theta}$

The proposed FL I system was added to the EKF which became FEKF that enhanced the filtering and reduced the estimation error of the tracking.

9. Dynamic System Model

System dynamic model that utilize all the heterogeneous sensors estimations and then fuse them is as following:

$$\tilde{x}_k = f(\tilde{x}_{k-1}, u_{k-1}) + w_{k-1} \quad (24)$$



$$\begin{bmatrix} \tilde{x}_k(1) \\ \tilde{x}_k(2) \\ \tilde{x}_k(3) \\ \tilde{x}_k(4) \\ \tilde{x}_k(5) \\ \tilde{x}_k(6) \end{bmatrix} = F * \begin{bmatrix} s_{k-1}^x \\ v_{k-1}^x \\ a_{k-1}^x \\ s_{k-1}^y \\ v_{k-1}^y \\ a_{k-1}^y \end{bmatrix} + Bu_{k-1} + \omega_{k-1} \quad (25)$$

Where s_k is the position, v_k is the velocity and a_k is the acceleration of the tracked agent at the current time estimated from the a priori measurements ($s_{k-1}, v_{k-1}, a_{k-1}$). ω_{k-1} refers to the system noise associated with each measured term. This model equation is applied for both x-axis and y-axis. B is the control vector matrix and u is the control vector that allow the control of the actuators in the system.

The velocity v_{k-1} and acceleration a_{k-1} first get integrated into position value. For the velocity to be calculated, the acceleration gets integrated into velocity value. The state transition matrix, that characterize the Jacobean matrix of nonlinear function ($\cdot$), can be written as:

$$F = \begin{bmatrix} 1 & dt & \frac{dt^2}{2} & 0 & 0 & 0 \\ 0 & 1 & dt & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & dt & \frac{dt^2}{2} \\ 0 & 0 & 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (26)$$

The process noise matrix is diagonal to all corresponding elements of the nonlinear system dynamic difference equation which result in:

$$Q = \begin{bmatrix} q_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & q_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & q_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & q_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & q_6 \end{bmatrix} \quad (27)$$

The measurement representation of the model for all states in the system is as follow:

$$\tilde{y}_k = h(\tilde{x}_{k-1}) + v_{k-1} \quad (28)$$

Where:

$$h = \begin{bmatrix} r \\ \theta \end{bmatrix} = \begin{bmatrix} \sqrt{(\tilde{x}_{k-1}(1))^2 + (\tilde{x}_{k-1}(4))^2} \\ \text{atan2}(\frac{\tilde{x}_{k-1}(4)}{\tilde{x}_{k-1}(1)}) \end{bmatrix} \quad (29)$$

r and θ are the polar coordinates representation of the distance and the angle between the reference point (sensor location in this case) and the tracked point (agent/object) which will then be transformed into Cartesian coordinates. Hence, the Jacobean matrix of h with respect to x is as following:



$$H = \begin{bmatrix} \frac{\tilde{x}_{k-1}(1)}{\sqrt{(\tilde{x}_{k-1}(1))^2 + (\tilde{x}_{k-1}(4))^2}} & 0 & 0 \\ \frac{-\tilde{x}_{k-1}(4)}{(\tilde{x}_{k-1}(1))^2 + (\tilde{x}_{k-1}(4))^2} & 0 & 0 \\ \frac{\tilde{x}_{k-1}(4)}{\sqrt{(\tilde{x}_{k-1}(1))^2 + (\tilde{x}_{k-1}(4))^2}} & 0 & 0 \\ 0 & \frac{\tilde{x}_{k-1}(1)}{(\tilde{x}_{k-1}(1))^2 + (\tilde{x}_{k-1}(4))^2} & 0 \end{bmatrix} \quad (30)$$

r and θ are correlated with different noises (measurement noises) which are not the same for all sensors taking in account the precision of each sensor:

$$R = \begin{bmatrix} r_{var} & 0 \\ 0 & \theta_{var} \end{bmatrix} \quad (31)$$

The proposed model diagram is shown below “Figure 8” which reveals how the data are collected first, then the features of all sensors is extracted and finally the decision making is made by hyper-fuzzy logic.



Figure 8: Monitoring System Model Block Diagram

10. Hyper-Fuzzy Decision

FL I only assumed that the systems are deterministic with no uncertainty which doesn't satisfy its own name “fuzzy” or the vagueness of human reasoning and decision-making that Zadeh substantiated his concepts and theories on. For that, in 1975 he extended FL I with other subgroups that can handle uncertainties in the system in three-dimensional structure. The new system was called Type-2 Fuzzy Logic system or Fuzzy Logic II (FL II) which is more complex and difficult to be implemented if compared with FL I [4].

In 2010, Salim proposed a new extension to the fuzzy set theory that combine the features of both FL I and FL II. The new controller was named Hyper-Fuzzy Logic (HF) which is based on the same membership functions of FL II on the top and the bottom but only apply FL I fuzzy sets [3,4].

Hyper fuzzy has various advantages since it is more advanced than FL I and easy to implement. First of all, it is easily adaptable. Specifically, one can learn how to apply it in various sectors without much problems. Another advantage is its immunity to noise. Another major advantage of HF is that it is considerably flexible. In fact, this system has included functions such as the capability to make human decisions. Furthermore, large quantities of data can be dealt with easily by this system. However, one

major setback is that HF has not yet been generalized on its own [4].

HF can be considered a special type of type-2 fuzzy sets. Unlike typical fuzzy, HF provides an allowance for imprecisions. As a matter of fact, it is derived by obtaining concepts of the upper and lower type-2 membership functions and is based on fuzzy sets from type 1 [3,4]. Furthermore, it is applied by utilizing a certain toolbox found in MATLAB software. Figure 11 shows how the HF membership function looks like as a blurred FL I membership function which was extended to contain uncertainty.

To utilize HF logic in this paper, the decisions of the security system are based on the suspiciousness degree (SD) of the moving agents around the monitored assets [4]. The number of moving agents (MaN), their corresponding distances from the assets around the monitored area (AdS), and the value of the asset (AsV) are the three inputs of the HF and the output will be a value between 0 and 12 (Figure 10) that indicates the suspiciousness degree (SD) of each agent at a certain time. The closer the agent to the asset, the more suspicious he/she would become; on the other hand, the farther the agent from the asset the less suspicious he/she is.

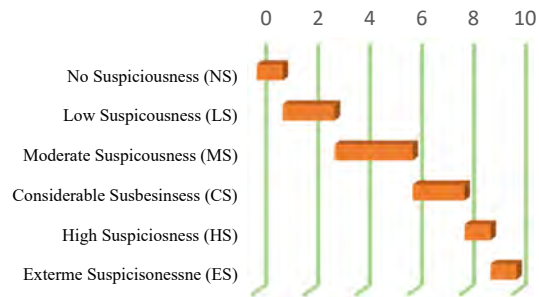


Figure 9: Suspiciousness Degree (SD)

There are six levels of SD, namely, No Suspiciousness (NS), Low Suspiciousness (LS), Moderate Suspiciousness (MS), Considerable Suspiciousness (CS), High Suspiciousness (HS), High Suspiciousness (HS), Extreme Suspiciousness (ES). Figure 9 explains the level of suspiciousness and their matching values.

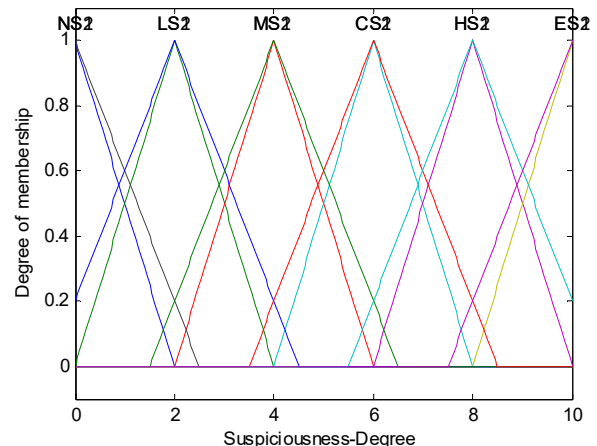


Figure 10: HF Membership function of SD



The moving agents corresponding distances from the assets (AdS) is classified as far (F), close (C), very close (VC), and enormously close (EC) (Figure 11).

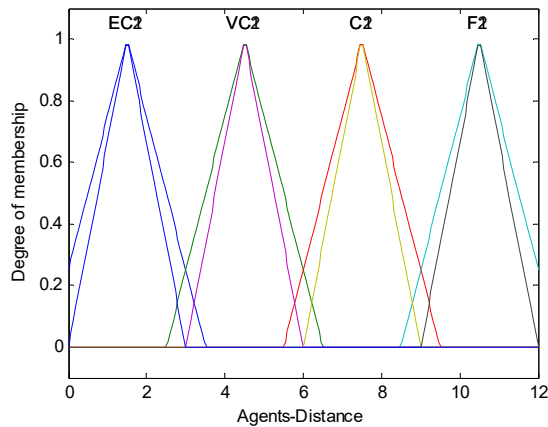


Figure 11: HF Membership function of AdS

The number of moving agents (MaN) is assumed to be variable; for that, they will be characterized as Rare (Ra), Average (Av), Moderate High (MH), and High (Hi) (Figure 12).

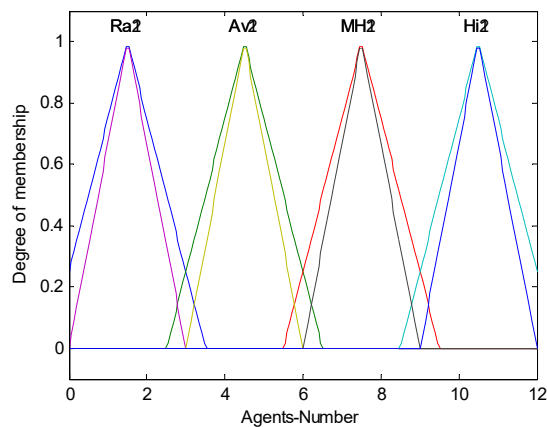


Figure 12: HF Membership function of MaN

The third input which is the value of the assets (AsV) has four main classes which are Not Expensive (NE), Average Expensive (AE), Expensive (E), and Too Expensive (TE) (Figure 13).

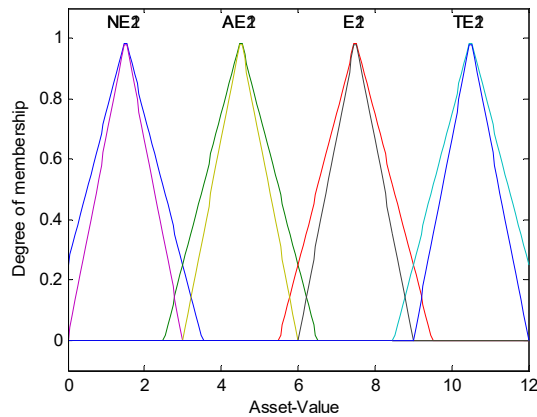


Figure 13: HF Membership function of AvS

The following figures shows the FIS of HF that controls and fine tune FEKF (Figure 14) and then the FIS surfaces

of AdS vs MaN and AvS vs MaN respectively (Figure 15 and Figure 16).

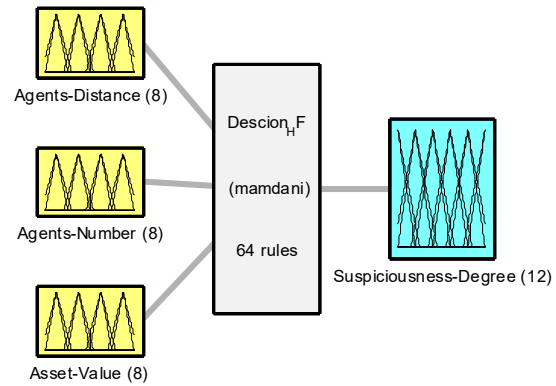


Figure 14: Decision Making HF logic decide the SD of each agent

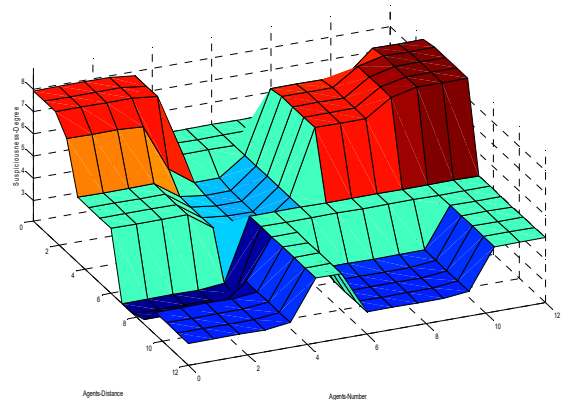


Figure 15: FIS Surface of SD with agents' number and distances as two inputs

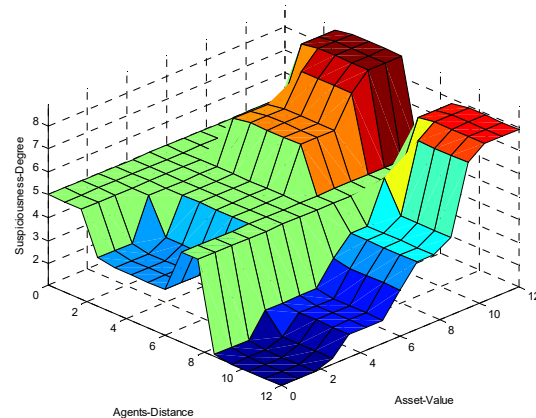


Figure 16: FIS Surface of SD with agents' distances and assets value as two inputs

11. Monitoring System Animation

Animating the monitoring environment including the moving agents and the sensors that track their movements in real-time is an effective addition that shows how every single component work together to emphasize the



functionality of the system elements, the sensors in action, and how the decision is made based on the collected data. Matlab based code and plots were created to simulate the proposed monitoring environment system. Sonars, IRs and Pressure sensor floor were designed carefully to track any moving object and how far or close are they from the monitored valuable assists. Three Sonar and IR arrays follow each agent while he/she is in covered area. The floor shows how the tiles interact with all agents showing their perspective locations and weights. Once the agent is detected initially with the pressure floor, the other sensors start to follow and feed the distance of the moving agent from the assets and his/her speed.

Figure 17 shows how the monitoring system is implemented and the location of all the sensors.

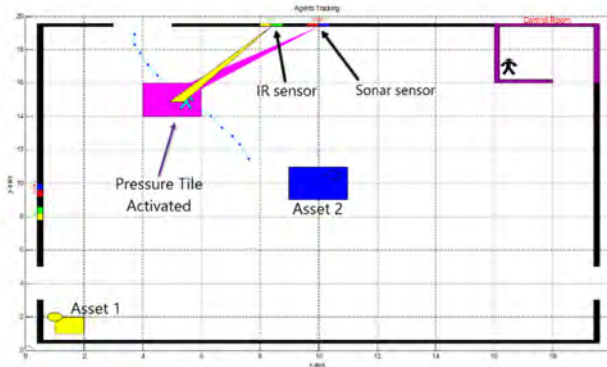


Figure 17: Animation of the monitoring system, sensors, assets and moving agents "Matlab"

12. Discussion and Future Work

FEKF provided a new extension of the EKF that allow to reduce the uncertainty in both the process and measurements under the FL supervision. Tuning the measurement noise covariance values R and process noise covariance Q showed how combining both FL and EKF can result in more stable estimations and reliable decisions consequently, FEKF namely.

Sensors similarity was performed utilizing MF to combine the homogeneous sensory data into a single data as well as their corresponding error covariances. Figure 18 provided below illustrate how the MF sensors within their groups can perform better.

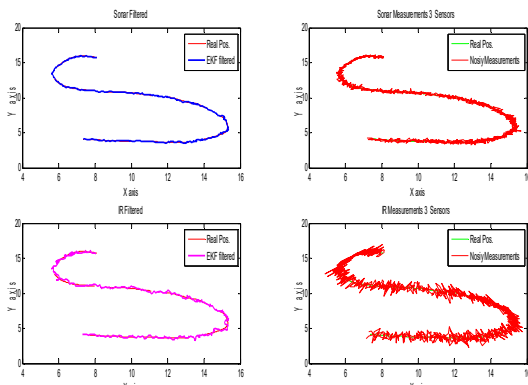


Figure 18: MF sonar and IR sensors

SVF was then applied to combine the resultant of measurement fusion inside the loop of iteration of FEKF which proved how the fused data can be more reliable and accurate if compared to the ordinary estimation method (Figure 19).

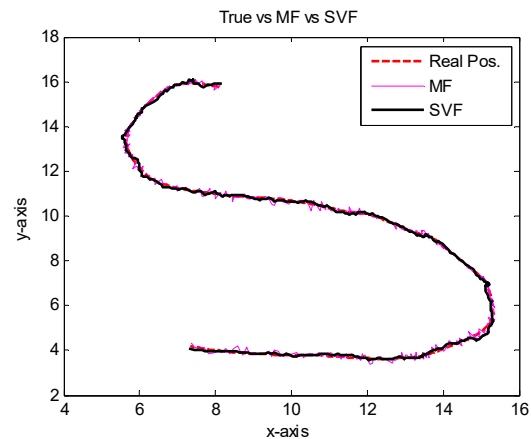


Figure 19: SVF compared to MF

The figure (Figure 20) below also shows how FEKF methodology can enhance the ordinary EKF where the mean square errors for both axis are compared.

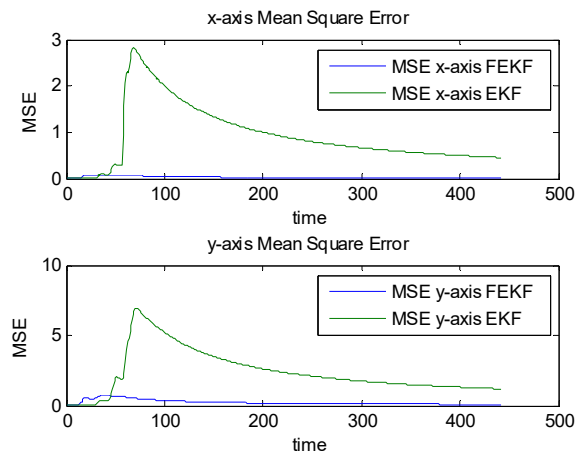


Figure 20: Mean Square Error FEKF vs EKF

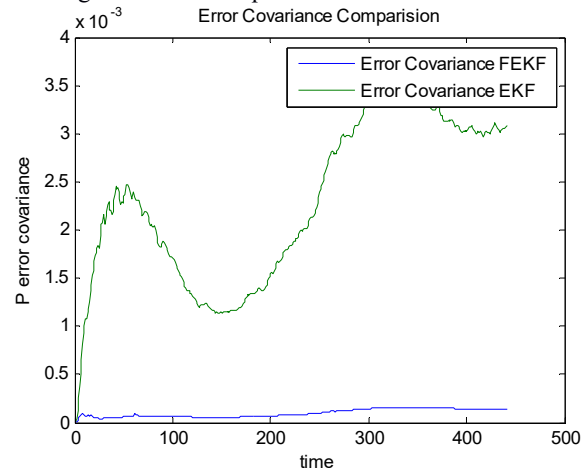


Figure 21: Mean Square Error FEKF vs EKF



Once the positions and weights are estimated for each agent, the data is sent to Hyper-Fuzzy Logic HFL to process their status and the degree of suspiciousness accordingly.

This monitoring system provides an advance security system that exceed many other systems utilizing unexpansive and affordable sensors sets.

The Matlab code for estimation and animation execution time was about 4.05 seconds for one agent moving more than 15 meters; however, with more sensors and agents' simulation the execution time of the code have increased slightly with an acceptable variance that allow the system to track agents and make decisions in real-time applications.

13. Conclusions

In this paper, a new security monitoring system applying data sensor fusion with the aid of Fuzzified extended Kalman filter have been developed and simulated. Sensors similarity and complementarity were the preprocessing methods for all sensors data which helped to feed the EKF with more accurate measurements and prevent the other uncertain readings from interfering during the filtering process. Fuzzified EKF sensor fusion has proven that multi-sensor security systems can outperform the systems which are based on single sensor measurements.

14. References

- [1] B. Moshiri, A. M. Khalkhali and H. R. Momeni, "Designing a home security system using sensor data fusion with DST and DSMT methods", 10th International Conference on Information Fusion, Quebec, pp. 1-6, 2007.
- [2] M. H. Shuvo and Y. Fu, "Sonar sensor virtualization for object detection and localization," SoutheastCon, Norfolk, VA, pp. 1-8, 2016.
- [3] T. K. Dakhallallah, M. A. Zohdy, O. M. Salim "Type-2 Fuzzy Kalman Hybrid Application for Dynamic Security Monitoring Systems based on Multiple Sensor Fusion", International Journal on Smart Sensing and Intelligent Systems, Vol.4, No.4, pp. 607-629, 2011.
- [4] O. M. Salim, M. A. Zohdy and H. S. Abdel-Aty-Zohdy, "Applications of hyper-fuzzy modeling and control for bio-inspired systems", 53rd IEEE International Midwest Symposium on Circuits and Systems, Seattle, WA, pp. 169-172, 2010.
- [5] L. Hall and J. Llinas, "An introduction to multisensor data fusion," in Proceedings of the IEEE, vol. 85, no. 1, pp. 6-23, Jan 1997.
- [6] J. D. de Jesus, J. J. V. Calvo and A. I. Fuente, "Surveillance system based on data fusion from image and acoustic array sensors," in IEEE Aerospace and Electronic Systems Magazine, vol. 15, no. 2, pp. 9-16, Feb 2000.
- [7] M. Taylor, R. Goubran and F. Knoefel, "Patient standing stability measurements using pressure sensitive floor sensors", IEEE International Instrumentation and Measurement Technology Conference Proceedings, Graz, pp. 1275-1279, 2012.
- [8] G. Welch, G. Bishop, "An Introduction to the Kalman Filter," UNCC Computer Science Technical Report 95041, 2005.
- [9] J.B. Gao, C.J. Harris, Some remarks on Kalman filters for the multisensor fusion, Information Fusion, Volume 3, Issue 3, Pages 191-201, ISSN 1566-2535, September 2002
- [10] H. F. Durrant-Whyte. Sensor Models and Multisensor Integration. International Journal of Robotics Research, 7(6):97-113, Dec. 1988.
- [11] John A. Fornshell and Alessandra Tesei, "The Development of SONAR as a Tool in Marine Biological Research in the Twentieth Century," International Journal of Oceanography, Article ID 678621, 9 pages, 2013.
- [12] Urlick, Robert J. Principles of Underwater Sound. Peninsula Pub, 2013.
- [13] Y. G. Kim, Y. Kim, S. H. Lee, S. T. Moon, M. Jeon and H. K. Kim, "Underwater acoustic sensor fault detection for passive sonar systems", First International Workshop on Sensing, Processing and Learning for Intelligent Machines (SPLINE), Aalborg, pp. 1-4, 2016.
- [14] W. Lo, K. L. Wu and J. S. Liu, "Wall following and human detection for mobile robot surveillance in indoor environment", IEEE International Conference on Mechatronics and Automation, Tianjin, pp. 1696-1702, 2014.
- [15] Habtie A.B., Abraham A., Midekso D. Comparing Measurement and State Vector Data Fusion Algorithms for Mobile Phone Tracking Using A-GPS and U-TDOA Measurements. In: Onieva E., Santos I., Osaba E., Quintián H., Corchado E. (eds) Hybrid Artificial Intelligent Systems. Lecture Notes in Computer Science, vol 9121. Springer, Cham, 2015.
- [16] H. Y. Wang, J. H. Park and Uk-Youl Huh, "Fuzzy-EKF for the mobile robot localization using ultrasonic satellite," 14th International Conference on Control, Automation and Systems (ICCAS), Seoul, pp. 1571-1575, 2014.
- [17] J. Simanek, M. Reinstein and V. Kubelka, "Evaluation of the EKF-Based Estimation Architectures for Data Fusion in Mobile Robots," in IEEE/ASME Transactions on Mechatronics, vol. 20, no. 2, pp. 985-990, April 2015.
- [18] J. Dong, D. Zhuang, Y. Huang, J. Fu, "Advances in Multi-Sensor Data Fusion: Algorithms and Applications", Sensors, 9, 7771-7784, 2009.
- [19] Hol, Jeroen. "Sensor Fusion and Calibration of Inertial Sensors, Vision, Ultra-wideband and GPS," Linköpings Universitet. Linköping, Sweden: Y LiU-Tryck. Dissertation No. 1368. Year 2011. ISBN 978-91-7393-197-7
- [20] Lundquist, Christian. "Sensor Fusion for Automotive Applications," Linköping University. Linköping, Sweden: LiU-Tryck, Dissertation No. 1409. Year 2011. ISBN 978-91-7393-023-9.



Biographies



Ali A. Alqudaihi was born in Dhahran, Saudi Arabia in 1983. He received his B. Sc. degree in Electrical Engineering from Saginaw Valley State University, Saginaw, Michigan in 2009. He received his M. Sc. degree in Electrical Engineering from Oakland University, Rochester Hills, Michigan in 2011. He is currently a Ph. D. student in Electrical Engineering at Oakland University, Rochester Hills, Michigan. His research interests are in the areas of security and monitoring systems, robotics, nonlinear control, nonlinear estimation and prediction, fuzzy logic and decision making.



Mohamed A. Zohdy was born in Caro, Egypt. He received B.Sc. degree in Electrical Engineering, Cairo University, Egypt in 1968 and he received his M. Sc. and Ph. D. degrees in Electrical Engineering from University of Waterloo, Canada in 1974, and 1977 respectively. He worked in various industries: dowty, iron and steel, and spar. He is currently a Professor at Oakland University, Rochester Hills, Michigan, U.S.A. His research interests are in areas of control, estimation, communication, neural networks, fuzzy logic and hybrid systems.



Hoda S. Abdel-Aty-Zohdy was born in Cairo, Egypt. She received B. Sc. degree (with Frist Class Honors), in Electrical Engineering, Cairo University, Egypt and she received her M. Sc. and Ph. D. degrees from University of Waterloo, Ontario, Canada, all in Electrical Engineering. She has founded the Microelectronics System Design Laboratory at Oakland University. She is currently a Professor at Oakland University, Rochester Hills, Michigan, U.S.A. Her research interests are in the areas of intelligent information processing, embedded Neural Networks, Genetic Algorithms, Analog ICs, Electronic Nose and Microelectronics. Dr. Abdel-Aty-Zohdy is a senior member for the Society of Women Engineers.

